

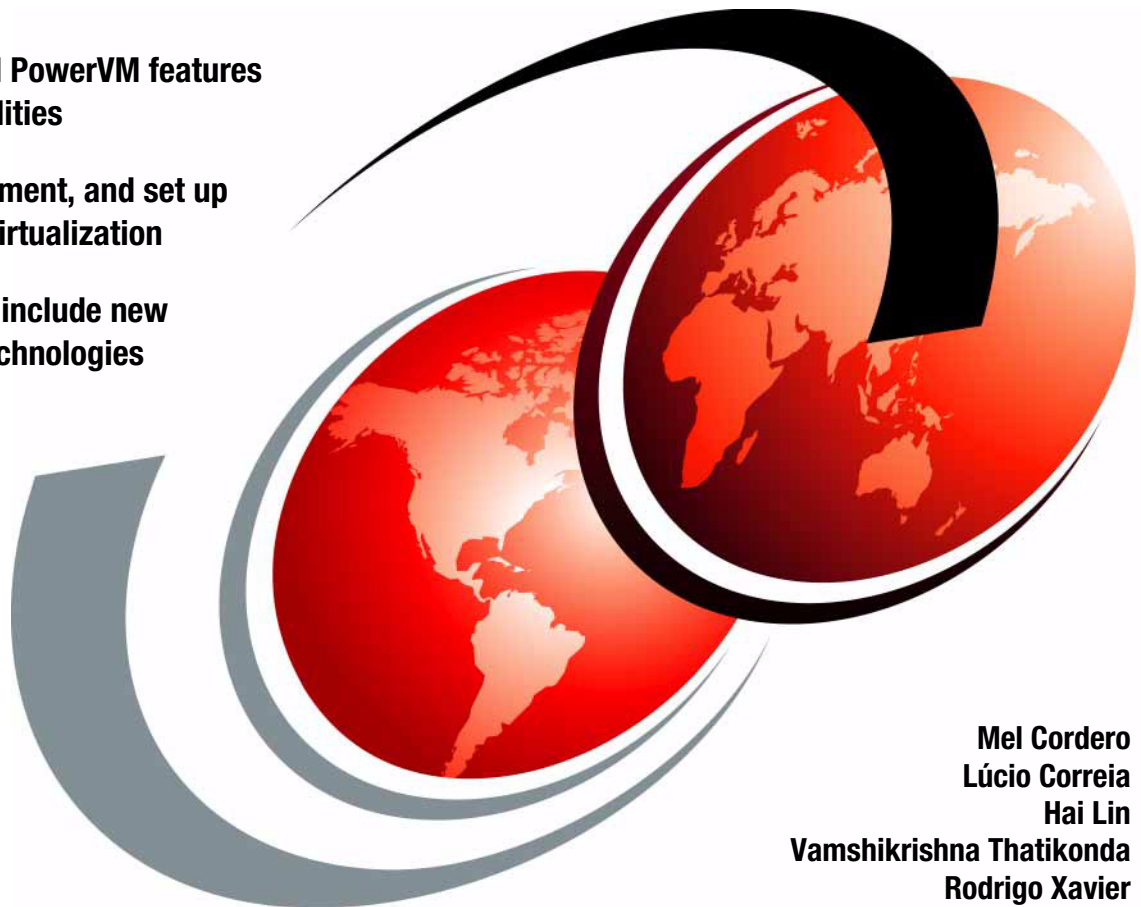


IBM PowerVM Virtualization Introduction and Configuration

Understand PowerVM features
and capabilities

Plan, implement, and set up
PowerVM virtualization

Updated to include new
POWER7 technologies



Mel Cordero
Lúcio Correia
Hai Lin

Vamshikrishna Thatikonda
Rodrigo Xavier



International Technical Support Organization

**IBM PowerVM Virtualization Introduction
and Configuration**

June 2013

Note: Before using this information and the product it supports, read the information in “Notices” on page xxi.

Sixth Edition (June 2013)

This edition applies to:

Version 7, Release 1 of AIX

Version 7, Release 1 of IBM i

Version 2, Release 2, Modification 2, Fixpack 26 of the Virtual I/O Server

Version 7, Release 7, Modification 6 of the HMC

Version AL730, release 95 of the POWER7 System Firmware

Version AL740, release 95 of the POWER7 System Firmware

© Copyright International Business Machines Corporation 2004, 2013. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	xi
Tables	xix
Notices	xxi
Trademarks	xxii
Preface	xxiii
Authors	xxiii
Now you can become a published author, too!	xxvi
Comments welcome	xxvi
Stay connected to IBM Redbooks	xxvii
Summary of changes	xxix
June 2013, Sixth Edition	xxix
Part 1. Overview	1
Chapter 1. PowerVM technologies	3
1.1 The value of PowerVM	4
1.2 What is PowerVM	4
1.2.1 New PowerVM version 2.2.2 features	6
1.2.2 PowerVM editions	7
1.2.3 Activating the PowerVM feature	12
1.3 The POWER Hypervisor	15
1.4 Logical partitioning technologies	17
1.4.1 Dedicated LPAR	17
1.4.2 Dynamic LPAR	17
1.4.3 Micro-partitioning	18
Chapter 2. Processor virtualization overview	19
2.1 Micro-partitioning	20
2.2 Shared-Processor Pools	21
2.3 Multiple Shared-Processor Pools	21
2.4 Shared dedicated capacity	23
Chapter 3. Memory virtualization overview	25
3.1 Active Memory Sharing overview	26
3.2 Active Memory Deduplication	27
3.3 Active Memory Expansion overview	29

3.4 Active Memory Mirroring overview	30
3.5 Memory virtualization technologies comparison	33
Chapter 4. I/O virtualization overview	35
4.1 Virtual I/O Server overview	37
4.1.1 Supported platforms	38
4.1.2 Virtual storage mapping	38
4.1.3 Virtual I/O Server network security	43
4.1.4 Command line and cfgassist interface	43
4.1.5 Hardware Management Console integration.	43
4.1.6 System Planning Tool support	44
4.1.7 Integrated Virtualization Manager	44
4.1.8 Tivoli support.	44
4.1.9 Allowed third party applications.	46
4.1.10 Performance Toolbox support.	46
4.1.11 Virtual I/O Server Performance advisor	47
4.2 Storage virtualization overview	47
4.2.1 Virtual SCSI.	48
4.2.2 Virtual Fibre Channel	50
4.2.3 Virtual Optical	52
4.2.4 Virtual Tape.	52
4.2.5 Virtual Console (virtual TTY/console support).	53
4.2.6 Availability	53
4.2.7 Shared storage pools	55
4.3 Network virtualization overview	61
4.3.1 Virtual Ethernet	62
4.3.2 Virtual Ethernet adapter	62
4.3.3 Virtual LAN	63
4.3.4 Virtual switches	64
4.3.5 Shared Ethernet Adapter	64
4.4 Platform consideration for I/O virtualization.	66
4.4.1 I/O virtualization consideration for IBM i	66
4.4.2 I/O virtualization consideration for Linux	74
Chapter 5. Server virtualization overview	79
5.1 Live Partition Mobility overview	80
5.2 Partition Suspend and Resume overview	82
Chapter 6. Management console overview	85
6.1 Management console comparison	86
6.2 Hardware Management Console.	86
6.3 Integrated Virtualization Manager	88
6.4 System Director VMControl.	88

Part 2. Plan	91
Chapter 7. PowerVM considerations	93
7.1 System requirements	94
7.1.1 Hardware requirements	94
7.1.2 Licensing requirements	97
7.1.3 Operating system requirements	98
7.2 Availability planning for PowerVM	100
7.2.1 Redundant Virtual I/O Servers	101
7.2.2 Live Partition Mobility	102
7.2.3 PowerVM with PowerHA	104
7.3 Security planning for PowerVM	105
7.4 Management Console considerations	107
7.4.1 Hardware Management Console (HMC)	107
7.4.2 Integrated Virtualization Manager (IVM)	108
7.4.3 IBM System Director VMControl	108
Chapter 8. Processor virtualization planning	111
8.1 Micro-partitioning capacity planning	112
8.1.1 Processing units of capacity	112
8.1.2 Capped and uncapped mode	113
8.1.3 Virtual processors	114
8.1.4 Shared processor considerations	117
8.2 Shared-Processor Pools capacity planning	119
8.2.1 Capacity attributes	119
8.2.2 The default Shared-Processor Pool (SPP ₀)	120
8.2.3 Server load and capacity examples	122
8.2.4 Shared-Processor Pools scenarios	127
8.3 Software license in a virtualized environment	133
8.3.1 Licensing factors in a virtualized system	134
8.3.2 Capacity capping	135
8.3.3 System with Capacity Upgrade on Demand processors	137
8.3.4 Summary of licensing factors	137
8.3.5 IBM software	138
8.3.6 Software licensing on IBM i	141
8.3.7 Linux operating system licensing	141
Chapter 9. Memory virtualization planning	143
9.1 Active Memory Sharing planning	144
9.1.1 Active Memory Sharing prerequisites	144
9.1.2 Deployment considerations	145
9.1.3 Sizing Active Memory Sharing	151
9.1.4 Usage examples	153
9.2 Active Memory Deduplication planning	157

9.2.1	Checking the requirements for Active Memory Deduplication	157
9.2.2	Sizing your systems for Active Memory Deduplication	165
Chapter 10.	I/O virtualization planning	167
10.1	Virtual I/O Server planning	168
10.1.1	Specifications required to create the Virtual I/O Server	168
10.1.2	Redundancy considerations	175
10.2	Storage virtualization planning	182
10.2.1	Virtual SCSI	182
10.2.2	Virtual Fibre Channel	189
10.2.3	Redundancy configurations for virtual Fibre Channel adapters . .	193
10.2.4	Virtual SCSI and Virtual Fibre Channel comparison	200
10.2.5	Virtual optical devices	203
10.2.6	Virtual tape devices	203
10.2.7	Availability planning for virtual storage	204
10.2.8	AIX LVM mirroring in the client partition	206
10.2.9	IBM i mirroring in the client partition	208
10.2.10	Linux mirroring in the client partition	208
10.2.11	Supported AIX client configurations	209
10.2.12	Supported virtual SCSI configurations	209
10.2.13	Shared storage pools	219
10.3	Network virtualization planning	223
10.3.1	Virtual Ethernet	223
10.3.2	Virtual LAN	223
10.3.3	Virtual switches	226
10.3.4	Shared Ethernet Adapter	229
10.3.5	Availability	236
10.3.6	Using Link Aggregation on the Virtual I/O Server	248
10.3.7	QoS	251
10.3.8	Performance considerations	253
Chapter 11.	Server virtualization planning	255
11.1	Dynamic LPAR operations and dynamic resources planning	256
11.1.1	Dedicated-processor partitions	256
11.1.2	Micro-partitions	256
11.1.3	Capacity on Demand	257
11.1.4	Planning for dynamic LPAR operations	258
11.1.5	Performing dynamic LPAR operations	260
11.2	Live Partition Mobility planning	260
11.2.1	General requirements	261
11.2.2	Migration capability and compatibility	262
11.2.3	Readiness	263
11.2.4	Migratability	263

11.2.5 Infrastructure	265
11.2.6 Virtual I/O Server	270
11.2.7 Live Partition Mobility using Virtual Fibre Channel	281
11.2.8 Multiple concurrent migrations	286
11.2.9 Remote Live Partition Mobility	288
11.2.10 Processor compatibility modes	292
11.2.11 General considerations	292
11.3 Suspend and Resume planning	296
11.3.1 Configuration requirements	296
11.3.2 The Reserved Storage Device Pool	298
11.3.3 Suspend/Resume and Shared Memory	301
11.3.4 Shutdown	306
11.3.5 Recover	307
11.3.6 Migrate	307
Part 3. Install	309
Chapter 12. I/O virtualization implementation	311
12.1 Creating a Virtual I/O Server	312
12.1.1 Creating the Virtual I/O Server partition	313
12.2 Installation of Virtual I/O Server	333
12.2.1 Changes on VIOS 2.2.1.0	335
12.2.2 Installing Virtual I/O Server using Optical device	337
12.2.3 Installing the Virtual I/O Server image using installios on HMC	342
12.2.4 Updating the Virtual I/O Server using fix packs	344
12.3 Defining virtual disks for client partitions	345
12.3.1 Defining virtual SCSI disks	345
12.3.2 Using file-backed devices	347
12.3.3 Using logical volumes	348
Chapter 13. Server virtualization implementation	353
13.1 Creating a client partition	354
13.1.1 Procedure	354
13.1.2 Dedicated donating processors	366
13.2 AIX client partition installation	368
13.3 Installation of IBM i client partition	375
13.3.1 Overview	375
13.3.2 Considerations for IBM i client partitions managed by IVM	376
13.3.3 Installation considerations	376
13.3.4 Installation process	376
13.4 Linux client partition installation	378
13.4.1 Unattended installations	378
13.4.2 Multipath devices and Linux installations	379
13.4.3 Starting a Linux installation from the network	379

13.4.4	Starting a Linux installation from Virtual Media Library	382
13.4.5	Installing IBM service and productivity tools	383
Part 4.	Set up	385
Chapter 14.	Processor virtualization setup	387
14.1	Configuring Multiple Shared-Processor Pools.	388
14.1.1	Dynamic adjustment of Maximum Pool Capacity	388
14.1.2	Dynamic adjustment of Reserve Pool Capacity	388
14.1.3	Dynamic movement between Shared-Processor Pools	389
14.1.4	Deleting a Shared-Processor Pool	389
14.1.5	Configuration scenario	389
14.2	Shared dedicated capacity	397
Chapter 15.	Memory virtualization setup	401
15.1	Active Memory Sharing setup	402
15.1.1	Creating the paging devices	402
15.1.2	Creating the shared memory pool.	404
15.1.3	Creating a shared memory partition	411
15.2	Active Memory Deduplication setup	414
15.2.1	Enabling AMD using the HMC GUI	414
15.2.2	Disabling AMD using the HMC GUI	416
15.2.3	Enabling or disabling AMD using the HMC CLI	417
Chapter 16.	I/O virtualization setup	419
16.1	Virtual I/O Server setup.	420
16.1.1	Command line interface	420
16.1.2	Mirroring the Virtual I/O Server rootvg	424
16.1.3	Virtual I/O Server security	427
16.2	Storage virtualization setup	466
16.2.1	Virtual SCSI.	466
16.2.2	Virtual Fibre Channel	474
16.2.3	Virtual optical.	490
16.2.4	Virtual tape	492
16.2.5	Availability	493
16.2.6	Availability configurations using multipathing	501
16.2.7	Availability configurations using mirroring	534
16.2.8	Shared storage pools	570
16.3	Network virtualization setup	580
16.3.1	Multiple VLANs	580
16.3.2	SEA failover	591
16.3.3	EtherChannel Backup in the AIX client	603
16.3.4	IBM i virtual IP address failover for virtual Ethernet adapters	608
16.3.5	Linux Ethernet connection bonding.	611

16.3.6	General rules for setting modes for QoS	615
16.3.7	Denial of Service hardening	615
Chapter 17.	Server virtualization setup	617
17.1	Live Partition Mobility setup	618
17.1.1	Live Partition Mobility enablement	618
17.1.2	Live Partition Mobility setup	621
17.1.3	Configuring the external storage	644
17.2	Suspend and Resume setup	648
17.2.1	Creating a reserved storage device pool	648
17.2.2	Creating a suspend and resume capable partition	655
17.2.3	Validating that a partition is suspend capable	658
17.2.4	Suspending a partition	660
17.2.5	Validating that a partition is resume capable	665
17.2.6	Resuming a partition	667
Part 5.	Appendix	671
Appendix A.	Recent PowerVM enhancements	673
A.1	New features in PowerVM2.2 and Virtual I/O Server Version 2.2 FP26	674
A.2	New features in PowerVM2.2 and Virtual I/O Server Version 2.2 FP25	675
A.3	New features in Version 2.2 FP24-SP1 of Virtual I/O Server	677
A.4	New features in Version 2.1 of Virtual I/O Server	678
Appendix B.	POWER processor modes	681
Appendix C.	Capacity on Demand	683
Appendix D.	Simultaneous Multithreading	685
D.1	POWER processor SMT	686
D.2	SMT and the operating system	687
D.2.1	SMT control in AIX	688
D.2.2	SMT control in IBM i	691
D.2.3	SMT control in Linux	692
Appendix E.	Active Memory Expansion	695
E.1	Prerequisites	696
E.2	Overview	696
E.3	Tools	698
E.4	Active Memory Expansion setup	699
Appendix F.	IBM i Virtual Partition Manager	703
F.1	Planning considerations	704
F.2	Creating an IBM i client partition using VPM	705
F.3	Creating a Linux partition using VPM	705

Appendix G. AIX Workload Partitions	707
G.1 Characteristics of WPARs	709
G.2 Types of WPARs	709
G.2.1 System WPARs	710
G.2.2 Application WPARs	710
G.3 Live Application Mobility	711
Appendix H. System Planning Tool	713
H.1 Sample scenario	714
H.2 Preparation recommendations	715
H.3 Planning the configuration with SPT	716
H.4 Initial setup checklist	724
Abbreviations and acronyms	727
Related publications	731
IBM Redbooks publications	731
Online resources	732
Help from IBM	736
Index	737

Figures

1-1	Example of virtualization activation codes website	13
1-2	HMC window to activate PowerVM feature	14
1-3	ASMI menu to enable the Virtualization Engine Technologies	15
1-4	POWER Hypervisor abstracts physical server hardware	16
2-1	Overview of the architecture of Multiple Shared-Processor Pools	22
3-1	Memory structure with Active Memory Deduplication enabled	28
3-2	A simple view of Active Memory Mirroring	30
3-3	A POWER7 processor book and its components	32
4-1	Simple Virtual I/O Server configuration	38
4-2	Virtual I/O Server concepts	40
4-3	Virtual I/O Server virtual Fibre Channel adapter mappings	51
4-4	Redundancy of Virtual SCSI using Dual Virtual I/O Server	54
4-5	Host bus adapter and Virtual I/O Server failover	55
4-6	Abstract image of the clustered Virtual I/O Servers	57
4-7	PowerVM Model with Shared Storage Pools	58
4-8	Thin-provisioned devices in the shared storage pool	60
4-9	Virtual Ethernet adapter reported on IBM i	68
4-10	Page conversion of 520-bytes to 512-bytes sectors	69
4-11	Virtual SCSI disk unit reported on IBM i	70
4-12	NPIV devices reported on IBM i	72
4-13	IBM i multipathing or mirroring for virtual SCSI	73
4-14	Single Virtual I/O Server with dual paths to the same disk	76
4-15	Dual Virtual I/O Server accessing the same disk	77
4-16	Implementing mirroring at client or server level	78
5-1	An Example for Live Partition Mobility	80
7-1	Sample for dual Virtual I/O Servers architecture	101
7-2	Migrating all partitions of a system	103
7-3	Live Partition Mobility capable virtualized environment diagram	104
8-1	Redistribution of ceded capacity within Shared-Processor Pool	122
8-2	Example of Multiple Shared-Processor Pools	123
8-3	Example of micro-partition moving between Shared-Processor Pools	125
8-4	The two levels of unused capacity redistribution	126
8-5	Example of a Web-facing deployment using Shared-Processor Pools	127
8-6	Web deployment using Shared-Processor Pools	128
8-7	Capped Shared-Processor Pool offering database services	130
8-8	Example of a system with Multiple Shared-Processor Pools	132
8-9	License boundaries with different processor and pool modes	138
8-10	Licensing requirements for a non-partitioned server	140

8-11 Licensing requirements in a micro-partitioned server	140
9-1 Non-overcommit	146
9-2 Logical overcommit	147
9-3 Physical overcommit	148
9-4 Est. additional CPU entitlement needed to support AMS paging device	152
9-5 Logical overcommitment of memory example.	155
9-6 Selecting Properties for a managed system in the HMC	160
9-7 Selecting Capabilities tab in HMC managed system Properties	160
9-8 Managed system capabilities in the HMC	161
9-9 IBM i DSPPTF command	163
9-10 IBM i DSPPTF command result	164
10-1 Redundant Virtual I/O Servers before maintenance	177
10-2 Redundant Virtual I/O Servers during maintenance	178
10-3 Separating disk and network traffic.	181
10-4 Setting maximum number of virtual adapters in a partition profile	186
10-5 Slot numbers that are identical in the source and target system	188
10-6 Comparing virtual SCSI and Virtual Fibre Channel.	189
10-7 Host bus adapter failover	193
10-8 Host bus adapter and Virtual I/O Server failover.	195
10-9 Heterogeneous multipathing configuration with Virtual Fibre Channel	196
10-10 Redundant Virtual I/O Server partitions with Virtual Fibre Channel	198
10-11 Virtual SCSI redundancy using multipathing and mirroring.	205
10-12 LVM mirroring with two storage subsystems.	207
10-13 Supported and best ways to mirror virtual disks	210
10-14 RAID5 configuration using a RAID adapter on the Virtual I/O Server.	211
10-15 Best way to mirror virtual disks with two Virtual I/O Server.	213
10-16 Using MPIO with IBM System Storage DS8000	215
10-17 Using MPIO on the Virtual I/O Server with IBM TotalStorage.	216
10-18 Configuration for IBM TotalStorage SAN Volume Controller	217
10-19 Configuration for multiple Virtual I/O Servers and IBM FAStT	218
10-20 Example of VLANs	224
10-21 Flow chart of virtual Ethernet	227
10-22 Shared Ethernet Adapter	230
10-23 VLAN configuration example.	232
10-24 Adding virtual Ethernet adapters on the Virtual I/O Server for VLANs	234
10-25 Basic SEA failover configuration.	238
10-26 Alternative configuration for SEA failover	241
10-27 Network redundancy using two Virtual I/O Servers and NIB.	242
10-28 SEA failover Primary-Backup configuration	244
10-29 SEA failover with Load Sharing.	245
10-30 Link Aggregation (EtherChannel) on the Virtual I/O Server	250
11-1 Network connections sample for Dynamic Logical Partitioning.	258
11-2 Set a proper Maximum virtual adapters value.	259

11-3	Hardware infrastructure enabled for Live Partition Mobility.	267
11-4	A mobile partition during migration	268
11-5	The final configuration after a migration is complete.	269
11-6	Enabling the Mover service partition MSP	271
11-7	Dual VIOS and client mirroring to dual VIOS before migration	274
11-8	Dual VIOS and client mirroring to dual VIOS after migration	275
11-9	Dual VIOS and client mirroring to single VIOS after migration	276
11-10	Dual VIOS and client multipath I/O to dual VIOS before migration	277
11-11	Dual VIOS and client multipath I/O to dual VIOS after migration	278
11-12	Single VIOS to dual VIOS before migration	279
11-13	Single VIOS to dual VIOS after migration	280
11-14	Basic virtual Fibre Channel infrastructure before migration	281
11-15	Basic virtual Fibre Channel infrastructure after migration	282
11-16	Dual VIOS and client multipath I/O to dual FC before migration.	284
11-17	Dual VIOS and client multipath I/O to dual VIOS after migration	285
11-18	Live Partition Mobility infrastructure with two HMCs	290
11-19	Live Partition Mobility infrastructure using private networks	291
11-20	One public and one private network migration infrastructure	292
11-21	Reserved Storage Device Pool.	300
11-22	Pool management interfaces	302
11-23	Shared Memory Pool and Reserved Storage Device Pool	303
12-1	Basic Virtual I/O Server scenario	312
12-2	Hardware Management Console server view	313
12-3	HMC Starting the Create Logical Partition wizard.	314
12-4	HMC Defining the partition ID and partition name.	315
12-5	HMC Naming the partition profile	316
12-6	HMC Select whether processors are to be shared or dedicated.	317
12-7	HMC Virtual I/O Server processor settings for a micro-partition	318
12-8	HMC Virtual I/O Server memory settings	320
12-9	HMC Virtual I/O Server physical I/O selection for the partition	322
12-10	HMC start menu for creating virtual adapters	324
12-11	HMC Selecting to create a virtual Ethernet adapter	325
12-12	HMC Creating the virtual Ethernet adapter	326
12-13	HMC Creating the virtual SCSI server adapter for the DVD	327
12-14	HMC Virtual SCSI server adapter for the NIM server	328
12-15	HMC List of created virtual adapters.	329
12-16	HMC Menu for creating Logical Host Ethernet Adapters	330
12-17	HMC Menu Optional Settings	331
12-18	HMC Menu Profile Summary	332
12-19	HMC The created partition VIO_Server1	333
12-20	HMC Activating a partition.	337
12-21	HMC Activate Logical Partition submenu	338
12-22	HMC Selecting the SMS menu for startup	339

12-23	The SMS startup menu	340
13-1	Creating client logical partition	354
13-2	Create Partition dialog.	355
13-3	The start menu for creating virtual adapters window	356
13-4	Creating a Client Ethernet adapter	357
13-5	Creating the client SCSI disk adapter	358
13-6	Creating the client SCSI DVD adapter	358
13-7	List of created virtual adapters	359
13-8	The Logical Host Ethernet Adapters menu	360
13-9	IBM i tagged I/O settings dialog	361
13-10	The Optional Settings menu	362
13-11	The Profile Summary menu	363
13-12	The list of partitions for the basic setup.	364
13-13	Backing up the profile definitions	365
13-14	The edit Managed Profile window.	366
13-15	Setting the Processor Sharing options	367
13-16	Activating the DB_server partition.	369
13-17	The SMS menu	370
13-18	Selecting the network adapter for remote IPL.	371
13-19	IP settings	372
13-20	Ping test	373
13-21	Setting the install device	374
13-22	IBM i Select load source device panel	377
14-1	Starting Shared-Processor Pool configuration	391
14-2	Virtual Shared-Processor Pool selection.	392
14-3	Shared-Processor Pool configuration	393
14-4	Virtual Shared-Processor Pool partition tab	394
14-5	Shared-Processor Pool partition assignment	394
14-6	Overview of Shared-Processor Pool assignments	395
14-7	The Edit Managed Profile window.	399
14-8	Setting the Processor Sharing options	400
15-1	Creating a shared memory pool	404
15-2	Defining the Pool size and Maximum pool size.	405
15-3	Selecting paging space partitions	406
15-4	Selecting paging devices	407
15-5	Selecting paging devices	409
15-6	Finishing shared memory pool creation	410
15-7	Defining a shared memory partition	411
15-8	Defining memory settings	413
15-9	Selecting a managed system in the HMC.	414
15-10	Opening the Shared Memory Pool Management window in the HMC	415
15-11	Enabling Active Memory Deduplication from HMC Pool Properties	416
15-12	Disabling Active Memory Deduplication from HMC Pool Properties.	416

16-1 Virtual I/O Server Config Assist Menu	420
16-2 The ikeyman program initial window	433
16-3 Create new key database window	434
16-4 Creating the ldap_server key	434
16-5 Setting the key database password	435
16-6 Default certificate authorities available on the ikeyman program	436
16-7 Creating a self-signed certificate initial panel	437
16-8 Self-signed certificate information	438
16-9 Default directory information tree created by mksecdap command	440
16-10 Starting the shared storage management HMC dialog	471
16-11 Creating a storage pool using the HMC	472
16-12 Defining storage pool attributes using the HMC GUI	473
16-13 Creating a virtual disk using the HMC	474
16-14 Virtual Fibre Channel adapter numbering	475
16-15 Dynamically add virtual adapter	477
16-16 Create Fibre Channel server adapter	478
16-17 Set virtual adapter ID	479
16-18 Save the Virtual I/O Server partition configuration	480
16-19 Change profile to add Virtual Fibre Channel client adapter	481
16-20 Create Fibre Channel client adapter	482
16-21 Define virtual adapter ID values	483
16-22 Select virtual Fibre Channel client adapter properties	485
16-23 Virtual Fibre Channel client adapter properties	486
16-24 IBM i logical hardware resources with Virtual Fibre Channel devices	488
16-25 SCSI setup for shared optical device	490
16-26 MPIO attributes	497
16-27 SAN attachment with multipathing across two Virtual I/O Servers	502
16-28 IBM i System Service Tools Display disk configuration status	517
16-29 IBM i System Service Tools Display disk unit details	518
16-30 IBM i client partition with added virtual SCSI adapter for multipathing	519
16-31 IBM i SST Display disk configuration status	522
16-32 IBM i SST Display disk path status	523
16-33 IBM i SST Display disk unit details	524
16-34 IBM i CPPEA33 message for a failed disk unit connection	525
16-35 IBM i SST Display disk path status after outage of Virtual I/O Server1	526
16-36 IBM i CPPEA35 message for a restored disk unit connection	527
16-37 IBM i SST Display disk path status	528
16-38 Linux client partition using MPIO to access SAN storage	529
16-39 Redundant Virtual I/O Server client mirroring scenario	534
16-40 VIO_Server2 physical adapter selection	535
16-41 Virtual SCSI adapters for VIO_Server2	537
16-42 IBM i SST Display disk configuration status	544
16-43 IBM i SST Display non-configured units	545

16-44	IBM i SST Display disk unit details	546
16-45	IBM i SST Specify ASPs to add units to	547
16-46	IBM i SST Problem Report Unit possibly configured for Power PC AS548	
16-47	IBM i SST Confirm Add Units	549
16-48	IBM i SST Selected units have been added successfully	550
16-49	IBM i partition restart to DST using a manual IPL	551
16-50	IBM i DST Enable remote load source mirroring.	552
16-51	IBM i DST Work with mirrored protection	553
16-52	IBM i DST Select ASP to start mirrored protection	554
16-53	IBM i DST Problem Report for Virtual disk units in the ASP	555
16-54	IBM i DST Virtual disk units in the ASP message	556
16-55	IBM i DST Confirm Start Mirrored Protection	557
16-56	IBM i Disk configuration information report	558
16-57	IBM i Licensed internal code IPL progress	559
16-58	IBM i Confirm Add Units	560
16-59	IBM i resulting mirroring configuration.	561
16-60	IBM i CPI0949 message for a failed disk unit connection	562
16-61	IBM i SST Display disk path status after outage of Virtual I/O Server1563	
16-62	IBM i CPI0988 message for resuming mirrored protection	564
16-63	IBM i SST Display disk configuration status for resuming mirroring . .	565
16-64	IBM i CPI0989 message for resumed mirrored protection	566
16-65	IBM i SST Display disk configuration status after resumed mirroring .	567
16-66	Linux client partition using mirroring with mdadm	568
16-67	Linux partitioning layout for mdadm mirroring	569
16-68	VLAN configuration scenario.	582
16-69	Virtual Ethernet configuration for the client partition using the HMC. .	583
16-70	Virtual Ethernet configuration for Virtual I/O Server using the HMC . .	585
16-71	HMC in a VLAN tagged environment	586
16-72	Cross-network VLAN tagging with a single HMC	588
16-73	Highly available Shared Ethernet Adapter setup	592
16-74	Create an IP on the Shared Ethernet Adapter using cfgassist	595
16-75	SEA failover Primary-Backup configuration	597
16-76	SEA failover with Load Sharing.	598
16-77	ECB configuration on AIX client	603
16-78	VIPA failover configuration for IBM i client	609
17-1	Network ping successful to remote HMC	621
17-2	HMC option for remote command execution.	622
17-3	Remote command execution window	623
17-4	Checking and changing LMB size with ASMI	624
17-5	Checking the amount of memory of the mobile partition.	625
17-6	Available memory on destination system	626
17-7	Checking the number of processing units of the mobile partition	628
17-8	Available processing units on destination system.	629

17-9	Synchronizing the time-of-day clocks	631
17-10	Virtual I/O Server Mover service partition property	633
17-11	Disable redundant error path handling	634
17-12	Verifying the number of serial adapters on the mobile partition	636
17-13	Disabling partition workload group - Other tab	637
17-14	Disabling partition workload group - Settings tab	638
17-15	Checking the number of BSR arrays on the mobile partition	639
17-16	Setting number of BSR arrays to zero	640
17-17	Checking if huge page memory equals zero	641
17-18	Setting Huge Page Memory to zero	643
17-19	IBM i partition property restricted IO setting	644
17-20	Checking free virtual slots	647
17-21	Reserved storage device pool management access menu	649
17-22	Reserved storage device pool management	650
17-23	Reserved storage device list selection	651
17-24	Reserved storage device selection	652
17-25	Reserved storage device pool creation	653
17-26	Creating a suspend and resume capable partition	655
17-27	Partition suspend menu	658
17-28	Validating suspend operation	659
17-29	Partition successful validation	659
17-30	Starting partition suspend operation	660
17-31	Running partition suspend operation	661
17-32	Finished partition suspend operation	662
17-33	Hardware Management Console suspended partition view	663
17-34	Reserved storage device pool properties	664
17-35	Partition resume menu	665
17-36	Validating resume operation	666
17-37	Successful validation	666
17-38	Starting partition resume operation	667
17-39	Running partition resume operation	668
17-40	Finished partition resume operation	669
17-41	Hardware Management Console resume view	670
D-1	Physical, virtual, and logical processors	687
D-2	SMIT SMT panel with options	690
D-3	IBM i processor multi-tasking system value	691
E-1	Active Memory Expansion example partition	697
E-2	Enabling Active Memory Expansion on the HMC	700
G-1	WPAR instantiated within dedicated partitions and micro-partitions	708
H-1	The partition and slot numbering plan of virtual storage adapters	714
H-2	The partition and slot numbering plan for virtual Ethernet adapters	715
H-3	The SPT Partition properties window	716
H-4	The SPT Virtual SCSI window	717

H-5 The SPT Edit Virtual Slots window	718
H-6 System Planning Tool ready to be deployed	719
H-7 Deploy System Plan	719
H-8 Deploy System Plan Wizard	719
H-9 The System Plan validation window	720
H-10 Partitions to Deploy	720
H-11 The Deployment Steps	721
H-12 The Deployment Progress window	722
H-13 Partition profiles deployed on the HMC	723

Tables

1-1	PowerVM features and technologies	4
1-2	Complementary technologies	5
1-3	Overview of PowerVM capabilities by edition	8
3-1	Memory virtualization comparison.	33
4-1	Kernel modules for IBM Power Systems virtual devices.	74
6-1	PowerVM Management console comparison	86
7-1	Virtualization features supported by POWER technology levels.	94
7-2	Server model to POWER technology level cross-reference	95
7-3	PowerVM feature code overview	97
7-4	Virtualization features supported by AIX, IBM i and Linux	98
7-5	PowerSC components, editions, and hardware support.	106
8-1	Reasonable settings for micro-partitions.	116
8-2	Entitled capacities for micro-partitions in a Shared-Processor Pool	120
8-3	Attribute values for the default Shared-Processor Pool (SPP0)	121
9-1	Active Memory Sharing requirements.	144
9-2	Estimated CPU entitlement requirements based on activity and storage	151
9-3	Prerequisites for Active Memory Deduplication	158
10-1	Resources that are required	168
10-2	Virtual I/O Server sizing examples	169
10-3	Virtual SCSI and Virtual Fibre Channel comparison	200
10-4	Inter-partition VLAN communication	233
10-5	VLAN communication to external network	235
10-6	Main differences between EC and LA aggregation.	249
10-7	Cap values for loose mode	253
13-1	IBM i client logical partition environments.	375
14-1	Micro-partition configuration and Shared-Processor Pool assignments	389
14-2	Shared-Processor Pool attributes	390
16-1	Default open ports on Virtual I/O Server	427
16-2	Hosts in the network	429
16-3	Task and associated command to manage Virtual I/O Server users . .	450
16-4	Authorizations corresponding to Virtual I/O Server commands	455
16-5	RBAC commands and their descriptions	463
16-6	Virtual SCSI adapter configuration for MPIO	504
16-7	Virtual SCSI adapter configuration for LVM mirroring.	536
16-8	Virtual Ethernet adapter overview for Virtual I/O Servers	593
16-9	EtherChannel Backup configuration examples	604
B-1	Differences between POWER6 and POWER7 modes.	682

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

Active Memory™	HACMP™	PowerLinux™
AIX®	i5/OS™	PowerVM®
AIX 5L™	IBM®	pSeries®
BladeCenter®	IBM PowerLinux™	Redbooks®
BNT®	iSeries®	Redpaper™
DB2®	OS/400®	Redbooks (logo)  ®
DS4000®	Parallel Sysplex®	Storwize®
DS6000™	Passport Advantage®	System Storage®
DS8000®	POWER®	SystemMirror®
Electronic Service Agent™	POWER Hypervisor™	Systems Director VMControl™
EnergyScale™	Power Systems™	Tivoli®
Enterprise Storage Server®	POWER6®	VMready®
eServer™	POWER6+™	Workload Partitions Manager™
GDPS®	POWER7®	XIV®
Geographically Dispersed Parallel Sysplex™	POWER7 Systems™	z/OS®
GPFS™	POWER7+™	
	PowerHA®	

The following terms are trademarks of other companies:

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

LTO, the LTO Logo and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Preface

This IBM® Redbooks® publication provides an introduction to IBM PowerVM® virtualization technologies on Power System servers.

PowerVM is a combination of hardware, firmware, and software that provides CPU, network, and disk virtualization. These are the main virtualization technologies:

- ▶ IBM POWER7®, IBM POWER6®, and POWER5 hardware
- ▶ POWER® Hypervisor
- ▶ Virtual I/O Server

Though the PowerVM brand includes partitioning, management software, and other offerings, this publication focuses on the virtualization technologies that are part of the PowerVM Standard and Enterprise Editions.

This publication is also designed to be an introduction guide for system administrators, providing instructions for these tasks:

- ▶ Configuration and creation of partitions and resources on the HMC
- ▶ Installation and configuration of the Virtual I/O Server
- ▶ Creation and installation of virtualized partitions
- ▶ Examples using IBM AIX®, IBM i, and Linux

This edition has been updated with the latest updates available and an improved content organization.

Authors

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

Mel Cordero is an IBM Senior IBM iSeries® Technical Specialist in IBM Australia since 2006. He has 25 years of experience in the Information Technology field. His areas of expertise include IBM i upgrade planning/implementation, work and security management, and system administration.

Lúcio Correia is a Staff Software Engineer in IBM Linux Technology Center Brazil. He has four years of experience on Linux software development and test on IBM Power Systems™. He holds a Masters degree in Distributed Systems and Networks from Universidade Federal de São Carlos (UFSCar). His areas of expertise include Virtual I/O Server and I/O Virtualization.

Hai Lin is a Complex Solution Manager from IBM China. He has eight years of experience in IBM pSeries®, Storage, and AIX post-sale support and services. He also has extensive experience in PowerVM, system virtualization, high availability, and disaster recovery solutions. He currently focuses on Managed Technical Support Service, which provides integrated solutions and helps clients to operate their IT environments better. He is an IBM eServer™ Certified Advanced Technical Expert - pSeries and IBM AIX 5L™.

Vamshikrishna Thatikonda is a Senior Staff Software Engineer in IBM Systems and Technology Group (India Software Labs), IBM India. He has over seven years of experience with IBM AIX Operating System Functional Testing and Virtual I/O Server Development. He holds a Bachelor degree in Computer Science and Information Technology from Jawaharlal Nehru Technological University, Hyderabad India. His areas of expertise include File Systems, Virtual I/O Server, PowerVM Virtualization, Shared Storage Pools.

Rodrigo Xavier has been working during the last six years at IBM Brazil (GTS Services Delivery) as a Subject Matter Expert, providing support to various customers. He has extensive experience with AIX and PowerVM in his daily job role, managing environments structured with IBM PowerVM on POWER6 and POWER7, especially features such as LPM. This includes starting on VIO Servers, client LPAR builds, hardware / LPAR migrations, and Live Partition Mobility. He holds a major degree in Computer Science, and is an IBM AIX Administrator and PowerVM Certified.

The project that produced this publication was managed by:

Scott Vetter, PMP. Scott is a Certified Executive Project Manager at the International Technical Support Organization, Austin Center. He has enjoyed 26 years of rich and diverse experience working for IBM in a variety of challenging roles. His latest efforts are directed at providing world-class Power Systems Redbooks publications, white papers, and workshop collateral.

Thanks to the following people for their contributions to this project:

David Bennin, Richard M. Conway, Ann Lund, Alfred Schwab
International Technical Support Organization, Poughkeepsie Center

Syed R Ahmed, Ray Anderson, John Banchy, Suman Batchu, Bob Battista, Ralph Baumann, Gail Belli, David Bennin, Thomas Bosworth, Ping Chen, Shaival Chokshi, Richard M. Conway, Joeseeph Czap, Herman Dierks, Arpana Durgaprasad, Linda Flanders, Nathan Fontenot, Chris Francois, Veena Ganti, Maria Garza, Djamel Ghaoui, Ron Gordon, Eric Haase, Robert Jennings, Yessong Brandon John, Brian King, Bob Kovacs, Monica Lemay, Yiwei Li, Chris Liebl, Ann Lund, Neal Marion, P Scott McCord, Francisco Moraes, Nidugala Muralikrishna, Steve Nasypany, Naresh Nayar,

Terrence Nixa, Jorge Noguerras, Paul F. Olsen, Jim Pafumi, Amartey Pearson,
Scott Prather, Ed Prosser, Michael Reed, Sergio Reyes, Joe Rinck,
Steven E Royer, Manash Sarma, Jeffrey Scheel, Ron Schmerbauch,
Alfred Schwab, Naoya Takizawa, Vi T. (Scott) Tran, Vasu Vallabhaneni,
Richard Wale, Steve Wallace, Robert Wallis, Duane Wenzel, Kristopher Whitney,
Joseph Writz, Bradley Vette, Laura Zaborowski
IBM US

Nigel Griffiths, Sam Moseley, Dai Williams
IBM UK

Joergen Berg
IBM Denmark

Bruno Blanchard
IBM France

Chanda A Sethia, Sangeeth Keeriyadath
IBM India

Thanks to the authors of the previous editions of this book.

- Authors of the first edition, *Advanced POWER Virtualization on IBM eServer p5 Servers: Introduction and Basic Configuration*, published in October 2004, were:

**Bill Adra, Annika Blank, Mariusz Gieparda, Joachim Haust,
Oliver Stadler, Doug Szerdi**

- Authors of the second edition, *Advanced POWER Virtualization on IBM System p5*, December 2005, were:

**Annika Blank, Paul Kiefer, Carlos Sallave Jr., Gerardo Valencia,
Jez Wain, Armin M. Warda**

- Authors of the third edition, *Advanced POWER Virtualization on IBM System p5: Introduction and Configuration*, February 2007, were:

Morten Vågmo, Peter Wüstefeld

- Authors of the fourth edition, *PowerVM Virtualization on IBM System p: Introduction and Configuration*, May 2008, were:

Christopher Hales, Chris Milsted, Oliver Stadler, Morten Vågmo

- Authors of the fifth edition, *PowerVM Virtualization: Introduction and Configuration*, June 2011, were:

**Stuart Devenish, Ingo Dimmer, Rafael Folco, Mark Roy, Stephane Saleur,
Oliver Stadler, Naoya Takizawa**

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

ibm.com/redbooks

- Send your comments in an email to:

redbooks@us.ibm.com

- Mail your comments to:

IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- ▶ Find us on Facebook:
<http://www.facebook.com/IBMRedbooks>
- ▶ Follow us on Twitter:
<http://twitter.com/ibmredbooks>
- ▶ Look for us on LinkedIn:
<http://www.linkedin.com/groups?home=&gid=2130806>
- ▶ Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:
<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>
- ▶ Stay current on recent Redbooks publications with RSS Feeds:
<http://www.redbooks.ibm.com/rss.html>

Summary of changes

This section describes the technical changes made in this edition of the book and in previous editions. This edition might also include minor corrections and editorial changes that are not identified.

Summary of Changes
for SG24-7940-05
for IBM PowerVM Virtualization Introduction and Configuration
as created or updated on March 20, 2014.

June 2013, Sixth Edition

This revision reflects the addition, deletion, or modification of new and changed information described below.

New information

- ▶ Capabilities provided by Virtual I/O Server Version 2, Release 2, Fixpack 26, including these:
 - Enhancements to Shared Storage Pools, see “Shared storage pools” on page 55
 - New VIOS Performance Advisor, see “Virtual I/O Server Performance advisor” on page 47
- ▶ Live Partition Mobility (LPM) improvements in PowerVM 2.2, see “Live Partition Mobility overview” on page 80, “Live Partition Mobility planning” on page 260 and “Live Partition Mobility setup” on page 618.
- ▶ IBM Active Memory™ Deduplication, see “Active Memory Deduplication” on page 27.
- ▶ Active Memory Mirroring, see “Active Memory Mirroring overview” on page 30.
- ▶ Power SC, see “Security planning for PowerVM” on page 105

Changed information

- ▶ The contents of the book have been reorganized into the following parts:
Overview, Plan, Install, Set Up.
- ▶ Several sections have been updated to include POWER7 based offerings.
- ▶ Sections describing the overview, plan, install, setup of Active Memory Deduplication, Active Memory Mirroring have been moved from the Redbooks publication, *Power Systems Memory Deduplication*, REDP-4827, *Power Systems Enterprise Servers with PowerVM Virtualization and RAS*, SG24-7965.
- ▶ Sections describing overview, plan, install, setup of Live Partition Mobility have been moved from the Redbooks publication, *IBM PowerVM Live Partition Mobility*, SG24-7460.



Part 1

Overview

Businesses are turning to PowerVM virtualization to consolidate multiple workloads onto fewer systems, increasing server utilization, and reducing cost. PowerVM technology provides a secure and scalable virtualization environment for AIX, IBM i, and Linux applications, built upon the advanced reliability, availability, and serviceability features and the leading performance of the Power Systems platform.

Part 1 of this publication provides an overview of IBM PowerVM concepts and technologies. It talks about what are PowerVM, PowerVM editions, PowerVM features, and how PowerVM virtualizes the processor, memory and I/O system.

This part includes the following topics:

- ▶ PowerVM technologies
- ▶ Processor virtualization overview
- ▶ Memory virtualization overview
- ▶ I/O virtualization overview
- ▶ Server virtualization overview
- ▶ Management console for PowerVM overview



PowerVM technologies

This chapter describes the value of PowerVM, PowerVM Editions, Hypervisor, and basic logical partitioning concepts.

It covers the following topics:

- ▶ The value of PowerVM
- ▶ What is PowerVM
- ▶ The POWER Hypervisor
- ▶ Logical partitioning technologies

1.1 The value of PowerVM

As you look for ways to maximize the return on your IT infrastructure investments, consolidating workloads becomes an attractive proposition.

IBM Power Systems combined with PowerVM technology are designed to help you consolidate and simplify your IT environment. Key capabilities include these:

- ▶ Improve server utilization and I/O resources sharing to reduce total cost of ownership and make better use of IT assets.
- ▶ Improve business responsiveness and operational speed by dynamically reallocating resources to applications as needed, to better match changing business needs or handle unexpected changes in demand.
- ▶ Simplify IT infrastructure management by making workloads independent of hardware resources, thereby enabling you to make business-driven policies to deliver resources based on time, cost and service-level requirements.

This chapter discusses the virtualization technologies and features on IBM Power Systems.

1.2 What is PowerVM

PowerVM provides the industrial-strength virtualization solution for IBM Power Systems servers and blades. This solution provides proven workload consolidation that helps clients control costs and improves overall performance, availability, flexibility and energy efficiency. PowerVM is a combination of hardware enablement and value-added software. Commonly, when we talk about PowerVM, we are talking about the features and technologies listed in Table 1-1.

Table 1-1 PowerVM features and technologies

Features and technologies	Function provided by
PowerVM Hypervisor	Hardware platform
Logical partitioning	Hypervisor
Micro-partitioning	Hypervisor
Dynamic logical partitioning	Hypervisor
Shared Processor Pools	Hypervisor
Integrated Virtualization Manager	Hypervisor, VIOS, IVM
Shared Storage Pools	Hypervisor, VIOS

Features and technologies	Function provided by
Virtual I/O Server	Hypervisor, VIOS
Virtual SCSI	Hypervisor, VIOS
Virtual Fibre Channel ^a	Hypervisor, VIOS
Virtual optical device & tape	Hypervisor, VIOS
Live Partition Mobility	Hypervisor, VIOS
Partition Suspend/Resume	Hypervisor, VIOS
Active Memory Sharing ^b	Hypervisor, VIOS
Active Memory Deduplication	Hypervisor
Active Memory Mirroring ^b	Hypervisor
Host Ethernet Adapter (HEA) ^c	Hypervisor

- a. Some other documents may call it as N_Port ID Virtualization (NPIV).
- b. Supported only by mid-tier and large-tier POWER7 Systems™ or later, including Power 770, 780, and 795.
- c. HEA is an hardware-based Ethernet virtualization technology used in IBM POWER6 and early POWER7 processor-based servers. Future hardware-based virtualization technologies will be based on Single Root I/O Virtualization (SR-IOV). For this reason, we do not discuss HEA configuration in this publication.

Also, the technologies in Table 1-2 are also frequently mentioned together with PowerVM. These technologies are covered in more detail in Part 5, “Appendix” on page 671.

Table 1-2 Complementary technologies

Features and technologies	Function provided by
POWER processor compatibility modes	Hypervisor
Capacity on Demand	Hypervisor
Simultaneous Multithreading	Hardware, AIX
Active Memory Expansion	Hardware ^a , AIX
AIX Workload Partitions	AIX ^b
System Planing Tool (SPT)	SPT

- a. Only available on POWER7 Systems and later
- b. Only available on AIX version 6.1 or later

PowerVM has three editions and each different edition has different features. In 1.2.2, “PowerVM editions” on page 7, we discuss the licensed features of each of the three different editions of PowerVM.

1.2.1 New PowerVM features

Power Systems servers coupled with PowerVM technology are designed to help clients build a dynamic infrastructure, reducing costs, managing risk, and improving service levels.

IBM PowerVM includes VIOS 2.2.2.1-FP26, HMC V7R7.6, and Power Systems firmware level 760. It contains the following enhancements for managing a PowerVM virtualization environment:

- ▶ Support for up to 20 partitions per processor, doubling the number of partitions supported per processor. This provides additional flexibility by reducing the minimum processor entitlement to 5% of a processor.
- ▶ Dynamic LPAR add or remove of virtual I/O adapters to or from a Virtual I/O Server partition:
 - The HMC V7R7.6 or later now automatically runs the add/remove commands (`cfgdev/rmdev`) on the Virtual I/O Server for the user. Prior to this enhancement, the user had to manually run these commands on the Virtual I/O Server.
- ▶ Ability for the user to specify the destination Fibre Channel port for any or all virtual Fibre Channel adapters.
- ▶ Improved Virtual I/O Server setup, tuning, and validation using the Runtime Expert.
- ▶ Live Partition Mobility supports up to 16 concurrent LPM activities.
- ▶ Shared Storage Pools create pools of storage for virtualized workloads, and can improve storage utilization, simplify administration, and reduce SAN infrastructure costs. The enhanced capabilities enable 16 nodes to participate in a Shared Storage Pool configuration, which can improve efficiency, agility, scalability, flexibility, and availability.

Shared Storage Pools flexibility and availability improvements include:

- IPv6 and VLAN tagging (IEEE 802.1Q) support for intermodal shared storage pools communication
- Cluster reliability and availability improvements
- Improved storage utilization statistics and reporting
- Nondisruptive rolling upgrades for applying service
- Advanced features that accelerate partition deployment, optimize storage utilization, and improve availability through automation

- ▶ New VIOS Performance Advisor analyzes Virtual I/O Server performance, and makes recommendations for performance optimization.
- ▶ PowerVM has the following new advanced features enabled by VMControl that accelerate partition deployment, optimize storage utilization, and improve availability through automation:
 - Linked clones allow for sharing of partition images, which greatly accelerates partition deployment and reduces the storage usage.
 - System pool management for IBM i workloads provides increased flexibility and resource utilization.
 - For further details about the appropriate IBM Systems Director VMControl™ release, visit this website:
<http://www.ibm.com/systems/software/director/vmcontrol/>

1.2.2 PowerVM editions

This section provides information about the virtualization capabilities of PowerVM. There are three versions of PowerVM, suited for various purposes:

- ▶ PowerVM Express Edition:
PowerVM Express Edition is designed for customers looking for an introduction to more advanced virtualization features at a highly affordable price.
- ▶ PowerVM Standard Edition:
PowerVM Standard Edition provides the most complete virtualization functionality for AIX, IBM i, and Linux operating systems in the industry. PowerVM Standard Edition is supported on Power Systems servers and includes features designed to allow businesses to increase system utilization.
- ▶ PowerVM Enterprise Edition:
PowerVM Enterprise Edition includes all the features of PowerVM Standard Edition plus two new industry-leading capabilities called Active Memory Sharing and Live Partition Mobility.

It is possible to upgrade from the Express Edition to the Standard or Enterprise Edition, and from Standard to Enterprise Editions. Table 1-3 outlines the functional elements of the available PowerVM editions.

Table 1-3 Overview of PowerVM capabilities by edition

PowerVM capability	PowerVM Express Edition	PowerVM Standard Edition	PowerVM Enterprise Edition
Maximum VMs	3 / Server	1000 / Server	1000 / Server
Micro-partitions ^a	Yes	Yes	Yes
Virtual I/O Server	Yes (Single)	Yes (Dual)	Yes (Dual)
Management	VMControl, IVM	VMControl, IVM ^b , HMC	VMControl, IVM ^b , HMC
Shared dedicated capacity	Yes	Yes	Yes
Multiple Shared-Processor Pools ^c	No	Yes	Yes
Live Partition Mobility	No	No	Yes
Active Memory Sharing ^d	No	No	Yes
Active Memory Deduplication ^d	No	No	Yes
Suspend/Resume	No	Yes	Yes
Virtual Fibre Channel	Yes	Yes	Yes
Shared Storage Pools	No	Yes	Yes
Thin provisioning	No	Yes	Yes
Thick provisioning	No	Yes	Yes

a. When the firmware is at level 7.6 or later, micro-partitions can be defined as small as 0.05 of a processor instead of 0.1 of a processor.

b. IVM only supports a single Virtual I/O Server

c. Needs IBM POWER6 processor-based system or later

d. Needs IBM POWER7 processor-based system with firmware at level 7.4 or later.

For an overview of the availability of the PowerVM features by Power Systems models, see this website:

<http://www.ibm.com/systems/power/software/virtualization/editions/features.html>

The PowerVM feature is a combination of hardware enablement and software that are available together as a single priced feature. It is charged at one unit for each activated processor, including software maintenance.

The software maintenance can be ordered for a one-year or three-year period. It is also charged for each active processor on the server.

When the hardware feature is specified with the initial system order, the firmware is shipped already activated to support the PowerVM features.

For an HMC-attached system with the PowerVM Standard Edition or the PowerVM Enterprise Edition, the processor-based license enables you to install several Virtual I/O Server partitions (usually two) on a single physical server to provide redundancy and to spread the I/O workload across several Virtual I/O Server partitions.

Virtual Ethernet and dedicated processor partition are features available without the PowerVM feature for servers attached to an HMC.

PowerVM Express Edition

The PowerVM Express Edition is designed for users looking for an introduction to more advanced virtualization features at a highly affordable price. It allows you to create up to three partitions per server. Partitions and the Virtual I/O Server are managed through the Integrated Virtualization Manager or VMControl with IBM System Director.

The PowerVM Express Edition provides the following capabilities:

Micro-partitioning support	Lets you create up to three partitions per server of minimum granularity of 1/20th of a CPU and a maximum of the entire server. Dynamically move processors and resources from one partition to another with no application downtime.
Integrated Virtualization Manager	Provides the capability to manage partitions and the Virtual I/O Server from a single point of control.
Shared Processor Pool	The Hypervisor automatically allocates processing power as needed by your applications according to the assigned priorities of partitions.
Virtual I/O Server	Provides virtual I/O resources to client partitions and enables shared access to physical I/O resource such as disks, tape, and optical media.

Virtual Fibre Channel	Provides direct access to Fibre Channel adapters from multiple client partitions, simplifying the management of Fibre Channel SAN environments.
Shared dedicated capacity	Allows the donation of spare CPU cycles for dedicated processor partitions to be utilized by the Shared Processor Pools, thus increasing overall system performance.

The Virtual I/O Server provides the IVM management interface for systems with the PowerVM Express Edition enabled. Virtual I/O Server is an appliance-style partition that is not intended to run end-user applications, and must only be used to provide login capability for system administrators.

PowerVM Standard Edition

The PowerVM Standard Edition includes features designed to allow businesses to increase system utilization while helping to ensure that applications continue to get the resources they need. Up to 1000 partitions can be created on larger IBM Power Systems.

Partitions: The maximum number of partitions per server depends on the server type and model. Details can be found at the following link:
<http://www.ibm.com/systems/power/hardware/reports/factsfeatures.html>

Compared to the PowerVM Express edition, the PowerVM Standard Edition additionally supports the following capabilities:

Hardware Management Console	Enables management of a set of IBM Power Systems from a single point of control.
Dual Virtual I/O Servers	Increases application availability by enabling Virtual I/O Server maintenance without a downtime for the client partitions.
Multiple Shared Processor Pools	Enables the definition of custom Shared Processor Pools to make allocation of CPU resource more flexible. This feature is supported on POWER6 and POWER7 Systems.
Shared Storage Pools	Provides distributed access to storage resources.

Thin provisioning

Enables more efficient provisioning of file-backed storage from a Shared Storage Pool by allowing the creation of file-backed devices that appear larger than the actual allocated physical disk space.

Suspend/Resume

Enables the saving of the partition state to a storage device from where the partition can later be resumed on the same server or on a different server.

PowerVM Enterprise

The PowerVM Enterprise Edition feature code enables the full range of virtualization capabilities that PowerVM provides. It allows users to exploit hardware resources in order to drive down costs and also provides maximum flexibility to optimize workloads across a server estate.

These are the primary additional capabilities in this edition:

- ▶ PowerVM Live Partition Mobility
- ▶ Active Memory Sharing

PowerVM Live Partition Mobility allows you to migrate running AIX, IBM i, and Linux partitions and their hosted applications from one physical server to another without disrupting the infrastructure services. The migration operation maintains complete system transactional integrity. The migration transfers the entire system environment, including processor state, memory, attached virtual devices, and connected users.

The benefits of PowerVM Live Partition Mobility include these:

- ▶ **Transparent maintenance:** This feature allows users and applications to continue operations by moving their running partitions to available alternative systems during the maintenance cycle.
- ▶ **Meeting increasingly stringent service-level agreements (SLAs):** This capability allows you to proactively move running partitions and applications from one server to another.
- ▶ **Balancing workloads and resources:** If a key application's resource requirements peak unexpectedly to a point where there is contention for server resources, you can move it to a larger server or move other, less critical, partitions to different servers, and use the freed-up resources to absorb the peak.

- Mechanism for dynamic server consolidation facilitating continuous server-estate optimization: Partitions with volatile resource requirements can use PowerVM Live Partition Mobility to consolidate partitions when appropriate or redistribute them to higher capacity servers at peak.

For more information about Live Partition Mobility, see the overview, planning, and setup parts of this book, 5.1, “Live Partition Mobility overview” on page 80, 11.2, “Live Partition Mobility planning” on page 260, and 17.1, “Live Partition Mobility setup” on page 618.

ACTIVE Memory Sharing is an IBM PowerVM advanced memory virtualization technology that provides system memory virtualization capabilities to IBM Power Systems, allowing multiple logical partitions to share a common pool of physical memory.

Active Memory Sharing can be exploited to increase memory utilization on the system either by decreasing the system memory requirement or by allowing the creation of additional logical partitions on an existing system. It also supports the Active Memory Deduplication feature.

For more information about Active Memory Sharing, see the planning and installing parts of this book, 9.1, “Active Memory Sharing planning” on page 144 and 15.1, “Active Memory Sharing setup” on page 402.

1.2.3 Activating the PowerVM feature

For upgrade orders, IBM will ship a key to enable the firmware, similar to the CUoD key. To find the current activation codes for a specific server, clients can visit the IBM website, where they can enter the machine type and serial number:

<http://www-912.ibm.com/pod/pod>

The activation code for PowerVM feature Standard Edition has a type definition of VET in the window results. You will see a window similar to that shown in Figure 1-1.

IBM Capacity on Demand: Activation code - Microsoft Internet Explorer

Address: <http://www-912.ibm.com/pod/pod>

Country/region [select] Terms of use

Search

Home | Products | Services & industry solutions | Support & downloads | My account

IBM Systems

Why IBM Systems

BladeCenter

Cluster servers

Mainframe

System i

System p

System x

UNIX

Solutions

Storage

Support

- Operating systems
- Alerts
- Developers
- Education
- Literature
- News and events

Related links

- Warranties, licenses and maintenance
- alphaWorks
- IBM Business Partners

Capacity on Demand

Activation code

To search for an activation code for a specific system, enter the information below and click **Submit**.

Search results

System Type: 9117 Serial Number: 10-1f170

Type	Activation Code	Posted Date (MM/DD/YYYY)
VET	458537EC15261717CA1F00002C2000411C	11/14/2007
VET	8ADB34FA4D005A4ACA1F00002C00004187	08/08/2007
POD	C541805856CC5A83567200000004004104	08/08/2007
MOD	9FE6BFE43452D3B45680000000320041E4	08/08/2007

Activation type definitions

POD: CUoD Processor Activation Code
MOD: CUoD Memory Activation Code
TCOD: On/Off CoD Enablement Code
TMOD: On/Off CoD Memory Enablement Code
VET: Virtualization Technology Code (PowerVM, Enterprise Enablement, WWP, Active Memory Expansion)
STDP: Standard Trial CoD Processor Activation Code
STDM: Standard Trial CoD Memory Activation Code
STME: Standard Trial Active Memory Expansion Code
EXCP: Exception Trial CoD Processor Activation Code
EXCM: Exception Trial CoD Memory Activation Code
USTA: Utility CoD Enablement Code
USTO: Utility CoD Termination Code
URPT: Utility CoD Reporting Code
PAID: Utility CoD PrePaid Code

Search for an activation code

System Type:

Serial Number: -

About IBM | Privacy | Contact

Figure 1-1 Example of virtualization activation codes website

For systems attached to an HMC, Figure 1-2 shows the HMC window where you activate the PowerVM feature. It may vary according to your HMC and system firmware level.

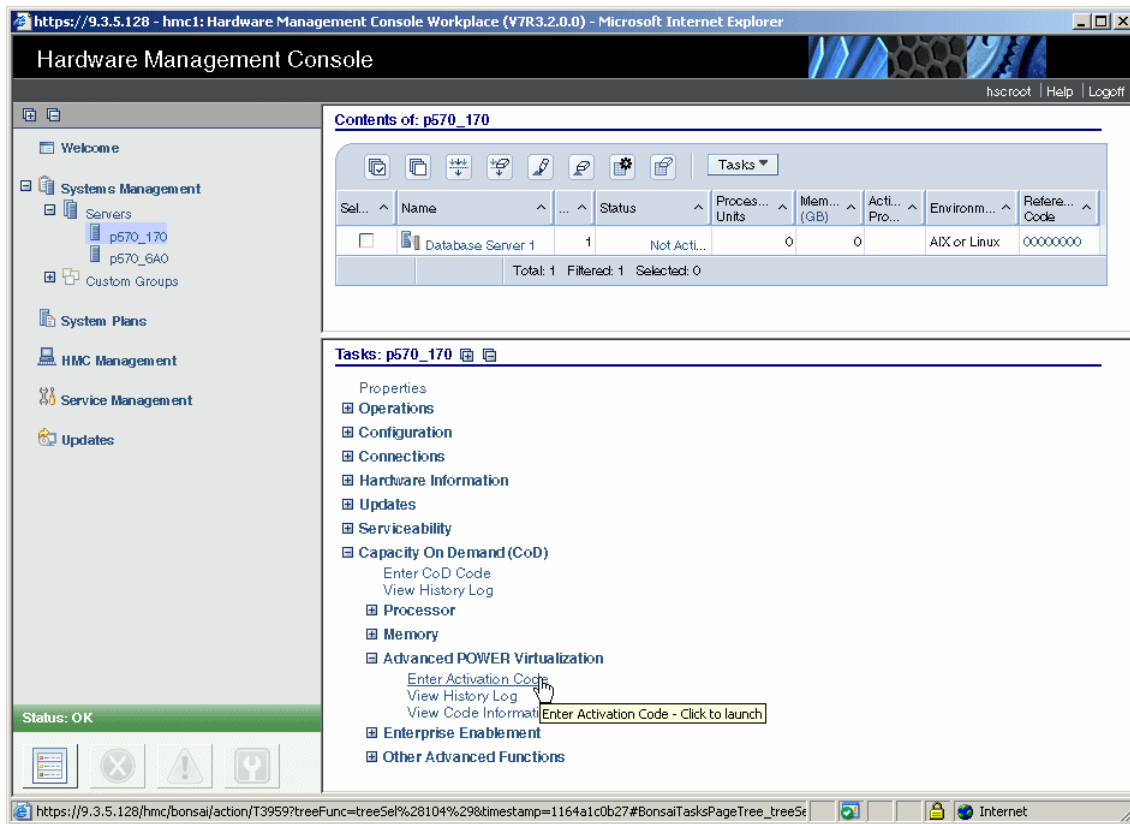


Figure 1-2 HMC window to activate PowerVM feature

When using the IVM within the Virtual I/O Server to manage a single system, Figure 1-3 shows the Advanced System Management Interface (ASMI) menu to enable the Virtualization Engine Technologies.

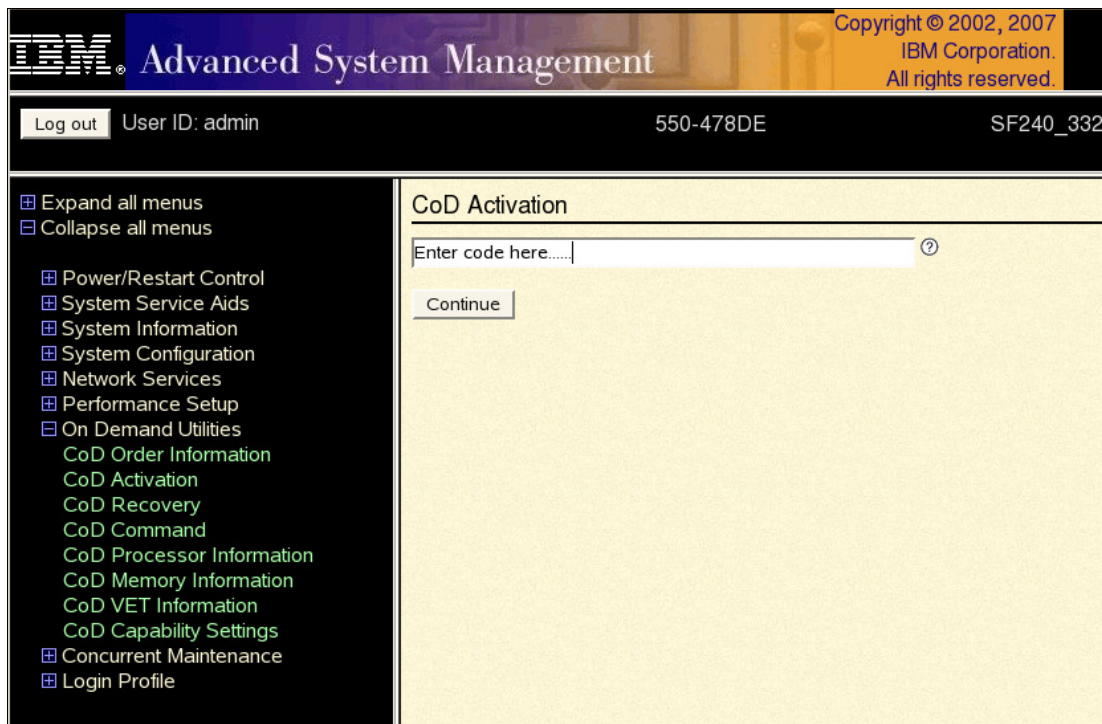


Figure 1-3 ASMI menu to enable the Virtualization Engine Technologies

1.3 The POWER Hypervisor

The IBM POWER Hypervisor™ is the foundation of IBM PowerVM. The POWER Hypervisor provides the ability to divide physical system resources into isolated logical partitions. Each logical partition operates like an independent system running its own operating environment: AIX, IBM i, Linux, or the Virtual I/O Server. The Hypervisor can assign dedicated processors, I/O, and memory, which you can dynamically reconfigure as needed, to each logical partition.

The Hypervisor can also assign shared processors to each logical partition using its micro-partitioning feature. Unknown to the logical partitions, the Hypervisor creates a Shared Processor Pool from which it allocates virtual processors to the logical partitions as needed. In other words, the Hypervisor creates virtual processors so that logical partitions can share the physical processors while running independent operating environments.

Combined with features designed into the IBM POWER processors, the POWER Hypervisor delivers functions that enable capabilities including dedicated-processor partitions, micro-partitioning, virtual processors, IEEE VLAN compatible virtual switch, virtual Ethernet adapters, virtual SCSI adapters, virtual Fibre Channel adapters, and virtual consoles.

The POWER Hypervisor is a firmware layer sitting between the hosted operating systems and the server hardware, as shown in Figure 1-4.

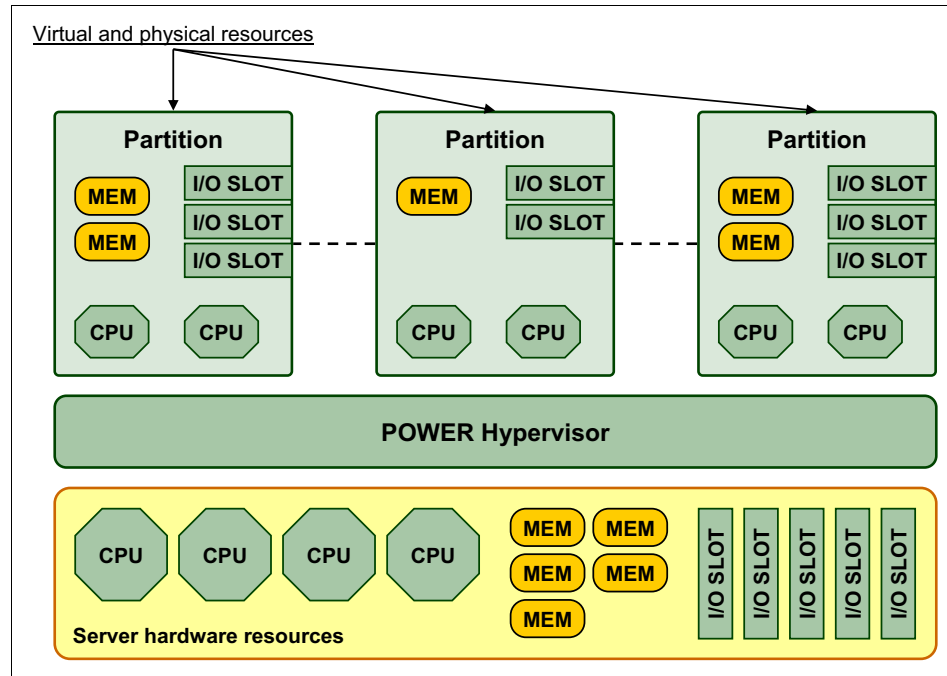


Figure 1-4 POWER Hypervisor abstracts physical server hardware

The POWER Hypervisor is always installed and activated, regardless of system configuration. The POWER Hypervisor has no specific or dedicated processor resources assigned to it.

The POWER Hypervisor performs the following tasks:

- ▶ Enforces partition integrity by providing a security layer between logical partitions.
- ▶ Provides an abstraction layer between the physical hardware resources and the logical partitions using them. It controls the dispatch of virtual processors to physical processors, and saves and restores all processor state information during virtual processor context switch.
- ▶ Controls hardware I/O interrupts and management facilities for partitions.

The POWER Hypervisor firmware and the hosted operating systems communicate with each other through POWER Hypervisor calls (**hcall**s).

1.4 Logical partitioning technologies

Logical partitions (LPARs) and virtualization increase utilization of system resources and add a new level of configuration possibilities. This section provides an overview on these topics.

1.4.1 Dedicated LPAR

Logical partitioning is available on all POWER5, POWER6 and POWER7 Systems or later. This technology offers the ability to make a server run as though it were two or more independent servers. When a physical system is logically partitioned, the resources on the server are divided into subsets called logical partitions (LPARs).

Processors, memory, and I/O devices can be individually assigned to logical partitions. The LPARs hold these resources for exclusive use. You can separately install and operate each dedicated LPAR because LPARs run as independent logical servers with the resources allocated to them. Since the resources are dedicated to use by the partition, it is called Dedicated LPAR.

1.4.2 Dynamic LPAR

Differently from a dedicated LPAR, dynamic logical partitioning (DLPAR) increased the flexibility, allowing selected system resources, such as processors, memory, and I/O components, to be added and deleted from logical partitions while they are executing. The ability to reconfigure dynamic LPARs encourages system administrators to dynamically redefine all available system resources to reach the optimum capacity for each defined dynamic LPAR.

For more information about dynamic logical partitioning, see 11.1, “Dynamic LPAR operations and dynamic resources planning” on page 256 and 17.1, “Live Partition Mobility setup” on page 618.

1.4.3 Micro-partitioning

Micro-partitioning technology allows you to allocate fractions of processors to a logical partition. A logical partition using fractions of processors is also known as a Shared Processor Partition or Micro-partition. Micro-partitions run over a set of processors called a Shared Processor Pool. Within the shared-processor pool, unused processor cycles can be automatically distributed to busy partitions as needed, which allows you to *right-size* partitions so that more efficient server utilization rates can be achieved. Implementing the shared-processor pool using micro-partitioning technology allows you to create more partitions on a server, which reduces costs.

Virtual processors are used to let the operating system manage the fractions of processing power assigned to the logical partition. From an operating system perspective, a virtual processor cannot be distinguished from a physical processor, unless the operating system has been enhanced to be made aware of the difference. Physical processors are abstracted into virtual processors that are available to partitions. The meaning of the term *physical processor* here is a *processor core*. For example, in a 2-core server there are two physical processors.

For more information about micro-partitioning, see 11.1.2, “Micro-partitions” on page 256.



Processor virtualization overview

The virtualization of physical processors in IBM Power Systems introduces an abstraction layer that is implemented within the IBM POWER Hypervisor. The POWER Hypervisor abstracts the physical processors and presents a set of virtual processors to the operating system within the micro-partitions on the system.

The operating system sees only the virtual processors and dispatches runnable tasks to them in the normal course of running a workload.

There are a range of technologies associated with processor virtualization.

This chapter covers the following topics:

- ▶ Micro-partitioning
- ▶ Shared-Processor Pools
- ▶ Multiple Shared-Processor Pools
- ▶ Shared-dedicated capacity

2.1 Micro-partitioning

Micro-partitioning is the ability to distribute the processing capacity of one or more physical processors among one or more logical partitions. Thus, processors are shared among logical partitions.

Micro-partitioning is supported across the entire POWER5 and later server range, from entry level to the high-end systems.

The benefit of micro-partitioning is that it allows significantly increased overall utilization of processor resources within the system. The micro-partition is provided with a processor entitlement—the processor capacity guaranteed to it by the POWER Hypervisor.

The granularity of processor entitlement is 0.01 of a processor with a required minimum of:

- ▶ 0.05 of processor for each micro-partition on IBM POWER7+™-based and later servers
- ▶ 0.10 of processor for each micro-partition on earlier POWER servers.

Thus processor entitlement can be precisely determined and configured. The micro-partitions share the available processing capacity, potentially giving rise to multiple partitions executing on the same physical processor.

When using Live Partition Mobility for partitions with 0.05 processor entitlement the destination server must also support 0.05 processor entitlement. If it does not then dynamic LPAR may be used to change the partition to 0.10 processor entitlement prior to using Live Partition Mobility.

Micro-partitions can use shared or dedicated memory. See Chapter 3, “Memory virtualization overview” on page 25 for more information.

The I/O requirements of a micro-partition can be supported through either physical and/or virtual resources. A micro-partition can own dedicated network and storage resources using dedicated physical adapters. Alternatively, micro-partitions might have some or all of the Ethernet and/or storage I/O resources satisfied through the use of virtual Ethernet, virtual SCSI, and virtual Fibre Channel.

2.2 Shared-Processor Pools

Micro-partitioning technology coupled with the POWER Hypervisor facilitates the sharing of processing units between micro-partitions. Processor resources can be used very efficiently with Shared-Process Pools (SPP), leading to a significantly increased overall system utilization.

A Shared-Processor Pool is a specific group of micro-partitions (and their associated virtual processors) through which processor capacity from the physical shared-processor pool can be managed.

The Physical Shared-Processor Pool is the set of physical processors that are not dedicated to any logical partition. All active physical processors are part of the Physical Shared-Processor Pool unless they are assigned to a dedicated-processor partition where:

- ▶ The logical partition is active and is not capable of capacity donation, or
- ▶ The logical partition is inactive (powered-off) and the systems administrator has chosen not to make the processors available for shared-processor work; see 14.2, “Shared dedicated capacity” on page 397.

The default Shared-Processor Pool (SPP₀)

On all Power Systems supporting Shared-Processor Pools, a default Shared-Processor Pool is always automatically defined. The default Shared-Processor Pool has a pool identifier of zero (SPP ID = 0) and can also be referred to as SPP₀.

The default Shared-Processor Pool has the same attributes as a user-defined Shared-Processor Pools except that these attributes are not directly under the control of the system administrator.

2.3 Multiple Shared-Processor Pools

Multiple Shared-Processor Pools (MSPPs) is a capability supported on POWER6 (or later) technology. This capability allows the user to define Shared-Processor Pools with the purpose of controlling the processor capacity that can be consumed from the Physical Shared-Processor Pool.

Multiple Shared-Processor Pools can be used to isolate workloads in a pool and thus not exceed an upper CPU limit. Multiple Shared-Processor Pools can also be useful for software license management where sub-capacity licensing is involved.

All Power Systems servers that support the Multiple Shared-Processor Pools capability will have a minimum of one (the default) Shared-Processor Pool and up to a maximum of 64 Shared-Processor Pools.

Micro-partitions are created and then identified as members of either the default Shared-Processor Pool (SPP₀) or a user-defined Shared-Processor Pool (SPP_n). The virtual processors that exist within the set of micro-partitions are monitored by the POWER Hypervisor and processor capacity is managed according to user-defined attributes.

An overview of the architecture of Multiple Shared-Processor Pools can be seen in Figure 2-1.

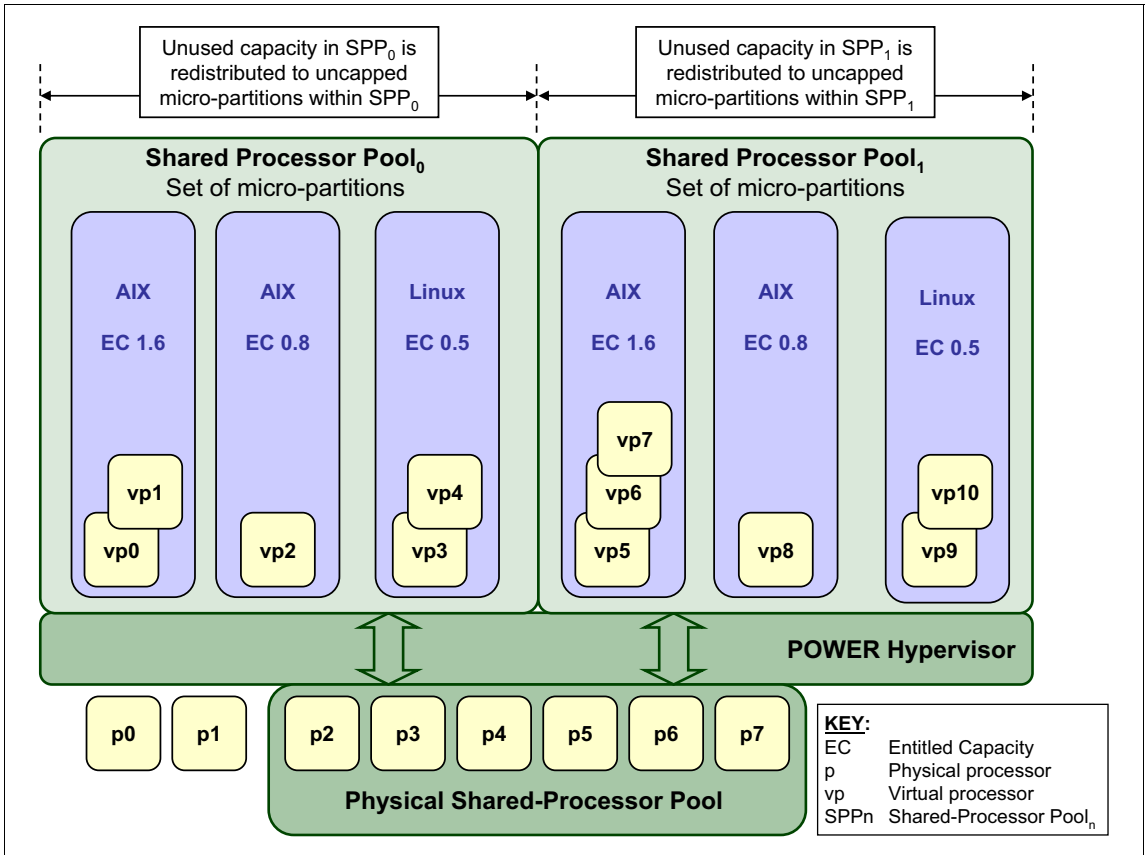


Figure 2-1 Overview of the architecture of Multiple Shared-Processor Pools

The diagram shows a system with eight processors (p0 to p7) and two Shared-Processor Pools (the default SPP₀ and the user-created SPP₁), with three micro-partitions each. Each micro-partition is seen by operating system as a individual system and may run a different AIX or Linux version.

The processors p0 and p1 are not donating capacity and, consequently, are not part of the Physical Shared-Processor Pool. The total capacity of processors p2 to p7 is distributed among the eleven virtual processors (vp0 to vp10) available on the Micro-Partitions. The POWER Hypervisor guarantees the minimum Entitled Capacity of each partition. Unused capacity in SPP₀ and SPP₁ is redistributed among their Micro-partitions.

For more details on micropartitioning and SPP capacity concepts, see Chapter 8, “Processor virtualization planning” on page 111.

Live Partition Mobility and Multiple Shared-Processor Pools

A micro-partition can leave a Shared-Processor Pool due to PowerVM Live Partition Mobility. Similarly, a micro-partition can join a Shared-Processor Pool in the same way. When performing PowerVM Live Partition Mobility, you are given the opportunity to designate a destination Shared-Processor Pool on the target server to receive and host the migrating Micro-partition. If a destination Shared-Processor Pool is not specified the default pool will be used.

Because several simultaneous micro-partition migrations are supported by PowerVM Live Partition Mobility, it is conceivable to migrate the entire Shared-Processor Pool from one server to another.

2.4 Shared dedicated capacity

POWER6-based and later servers offer the capability of harvesting unused processor cycles from dedicated-processor partitions. These unused cycles are then donated to the physical Shared-Processor Pool associated with micropartitioning. This ensures the opportunity for maximum processor utilization throughout the system.



Memory virtualization overview

POWER technology-based servers are very powerful and provide a lot of processor capacity. Memory is therefore often the bottleneck that prevents an increase in the overall server utilization.

This chapter describes the memory virtualization capabilities available for PowerVM:

- ▶ Active Memory Sharing overview
- ▶ Active Memory Deduplication overview
- ▶ Active Memory Expansion overview
- ▶ Active Memory Mirroring overview
- ▶ Memory virtualization technologies comparison

3.1 Active Memory Sharing overview

Active Memory Sharing is an IBM PowerVM advanced memory virtualization technology that provides system memory virtualization capabilities to IBM Power Systems, allowing multiple partitions to share a common pool of physical memory.

Active Memory Sharing is only available with the PowerVM Enterprise edition.

The physical memory of a IBM Power System server can be assigned to multiple partitions either in a dedicated mode or a shared mode. The system administrator has the capability to assign part of the physical memory to a partition and other physical memory to a pool that is shared by other partitions. A single partition can have either dedicated or shared memory.

- ▶ With a pure dedicated memory model, it is the system administrator's task to optimize available memory distribution among partitions. When a partition suffers degradation due to memory constraints and other partitions have unused memory, the administrator can react manually by issuing a dynamic memory reconfiguration.
- ▶ With a shared memory model, it is the system that automatically decides the optimal distribution of the physical memory to partitions and adjusts the memory assignment based on partition load. The administrator reserves physical memory for the shared memory pool, assigns partitions to the pool, and provides access limits to the pool.

Active Memory Sharing can be exploited to increase memory utilization on the system either by decreasing the global memory requirement or by allowing the creation of additional partitions on an existing system. Active Memory Sharing can be used in parallel with Active Memory Expansion on a system running a mixed workload of various operating systems. For example, AIX partitions can take advantage of Active Memory Expansion while other operating systems take advantage of Active Memory Sharing.

You can also go to the information center by the following link:

http://pic.dhe.ibm.com/infocenter/powersys/v3r1m5/topic/p7eew/p7eew_ams.htm

3.2 Active Memory Deduplication

Active Memory Deduplication is a PowerVM technology that minimizes the existence of identical memory pages in main memory space. When running workloads on traditional virtual LPARs, multiple identical data are saved across different positions in main memory. This is quite common with memory pages containing code instructions.

To optimize memory use, Active Memory Deduplication avoids data duplication in multiple distinct memory spaces. Active Memory Deduplication coalesces the data in just one physical memory page and frees the other chunks with identical data. The result is multiple logical memory pages pointing to the same physical memory page, thus saving memory space.

Active Memory Deduplication only works with partitions configured with shared memory. That is, to enable Active Memory Deduplication in your system, you must configure your LPARs to use Active Memory Sharing. The goal of Active Memory Sharing is to share memory among multiple LPARs of a system and therefore increase server consolidation by creating memory overcommitment.

When the aggregate working memory set size of the partitions exceeds the shared pool size, disk I/O occurs. This condition is called *physical memory overcommitment*, but it does not frequently happen. Nevertheless, a way to minimize these disk operations is to reduce the amount of physical memory use, thus reducing the probability of physical memory overcommitment. Active Memory Deduplication achieves this goal, thus enhancing the performance of an Active Memory Sharing enabled environment.

To deduplicate memory, PowerVM changes its logical memory map to make all identical logical memory pages point to the same physical memory page, as shown in Figure 3-1. In this figure, the pages that contain the same data are shown in blue.

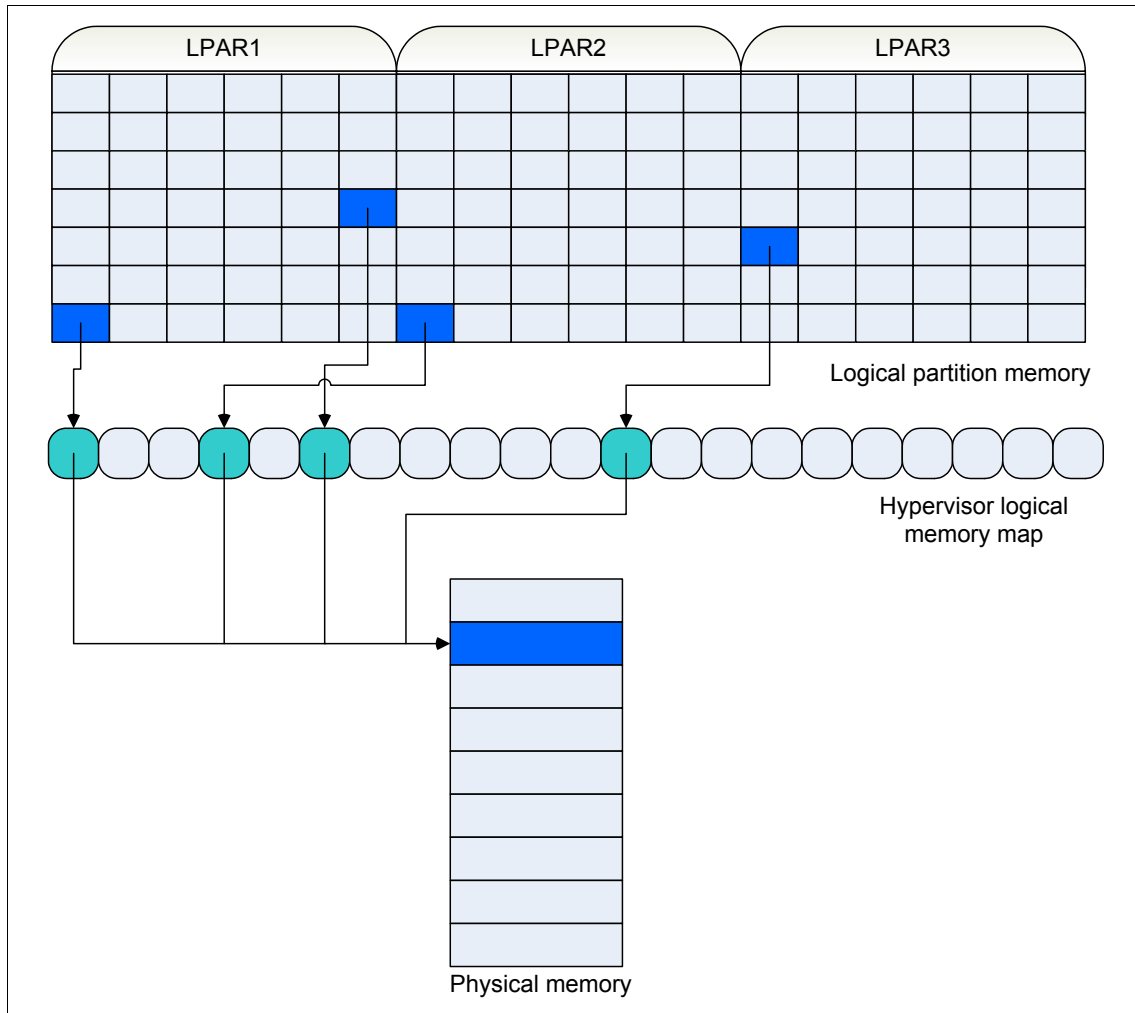


Figure 3-1 Memory structure with Active Memory Deduplication enabled

The memory savings obtained through the use of Active Memory Deduplication returns to the shared memory pool and is made available for reuse by the LPARs within the memory pool.

The latest enhancements for PowerVM V2.2 include Active Memory Deduplication detection and removal of duplicate memory pages to optimize memory usage in an Active Memory Sharing environment.

3.3 Active Memory Expansion overview

Active Memory Expansion is an innovative POWER7 (or later) technology that allows the effective maximum memory capacity to be much larger than the true physical memory maximum. Compression and decompression of memory content can allow memory expansion up to 100%. This can allow a partition to do significantly more work or support more users with the same physical amount of memory. Similarly, it can allow a server to run more partitions and do more work for the same physical amount of memory.

Active Memory Expansion is not a PowerVM feature. It relies on POWER7 (or later) hardware and AIX technologies. But since it is often mentioned together with other PowerVM memory virtualization technologies, this publication also talks some about it to help your understand them better.

Active Memory Expansion uses CPU resources of a partition to compress or decompress the memory contents of this same partition. The trade off of memory capacity for processor cycles can be an excellent choice, but the degree of expansion varies based on how compressible the memory content is, and it also depends on having adequate spare CPU capacity available for the compression or decompression. Tests in IBM laboratories using sample work loads showed excellent results for many workloads in terms of memory expansion per additional CPU utilized. Other test workloads had more modest results.

Active Memory Expansion Enablement is an optional hardware feature of POWER7 (or later) offerings. You can order this feature when initially ordering the server, or it can be purchased later.

Latest IBM Power Systems October 2012 announcement on POWER7+ hardware accelerator for Active Memory Expansion delivers 25% higher levels of memory expansion than available with POWER7 processor chips. While POWER7 systems offer up to 100% memory expansion which can effectively double the server's maximum memory, POWER7+ server offer up to 125% memory expansion for AIX partitions. Thus, a system memory maximum of 4 TB could effectively become 9 TB effective memory capacity.

For additional information regarding Active Memory Expansion, you can read Appendix E, "Active Memory Expansion" on page 695.

3.4 Active Memory Mirroring overview

Active Memory Mirroring for the hypervisor is a RAS function that is provided with POWER7 high-end systems such as Power 795, Power 780 and Power770.

This feature is also sometimes referred to as *system firmware mirroring*. Do not confuse it with other memory technologies, such as Active Memory Sharing and Active Memory Expansion, which are discussed in 3.1, “Active Memory Sharing overview” on page 26 and 3.3, “Active Memory Expansion overview” on page 29.

Active Memory Mirroring for the hypervisor is designed to mirror the main memory that is used by the system firmware to ensure greater memory availability by performing advance error-checking functions. This level of sophistication in memory reliability on Power systems translates into outstanding business value. When enabled, an uncorrectable error that results from a failure of main memory used by the system firmware will not cause a system-wide outage. The system maintains two identical copies of the system hypervisor in memory at all times, as shown in Figure 3-2.

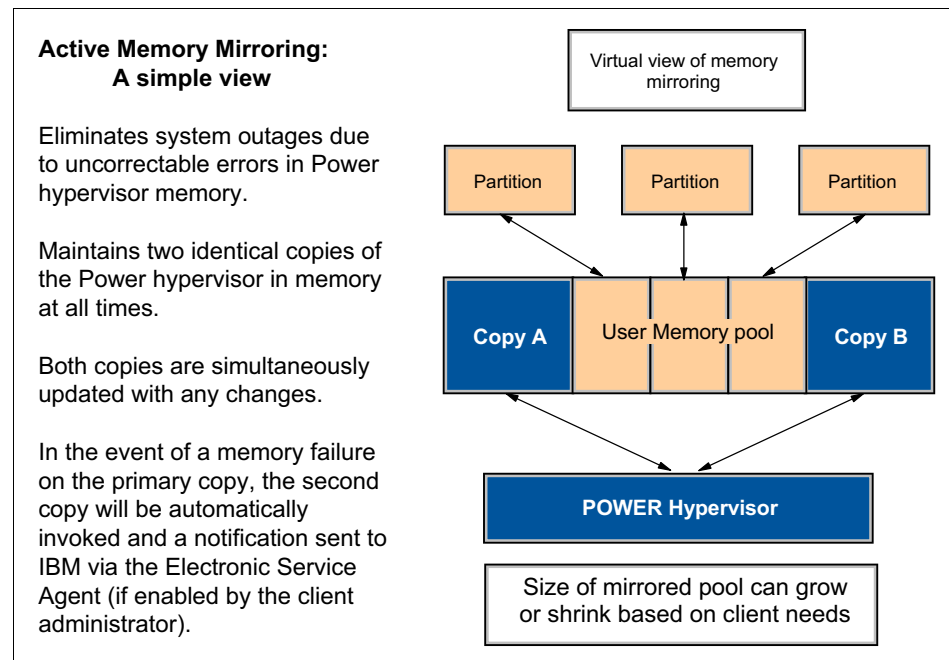


Figure 3-2 A simple view of Active Memory Mirroring

When a failure occurs on the primary copy of memory, the second copy is automatically invoked and a notification is sent to IBM via the IBM Electronic Service Agent™ (ESA). Implementing the Active Memory Mirroring function requires additional memory; therefore, you must consider this requirement when designing your server.

Depending on the system I/O and partition configuration, between 5% and 15% of the total system memory is used by hypervisor functions on a system on which Active Memory Mirroring is not being used. Use of Active Memory Mirroring for the hypervisor doubles the amount of memory that is used by the hypervisor, so appropriate memory planning must be performed. The System Planning Tool (SPT) can help estimate the amount of memory that is required. See Chapter 4 “Planning for virtualization and RAS in POWER7 high-end servers” in *Power Systems Enterprise Servers with PowerVM Virtualization and RAS*, SG24-7965 for more details.

Active Memory Mirroring for the hypervisor is provided as part of the hypervisor, so there is no feature code that needs to be ordered that provides this function. It is also mandatory that the server has enough free memory to accommodate the mirrored memory pages.

Active Memory Mirroring is required and automatically enabled on the Power 780 and on the Power 795. However, on the Power 770 AMM is optional and is ordered and enabled via feature #4797. An optimization tool for memory defragmentation is also included as part of the Active Memory Mirroring feature.

Disabling Active Memory Mirroring: Active Memory Mirroring can be disabled on a system if required, but you must remember that disabling this feature leaves your Power server exposed to possible memory failures that can result in a system-wide outage.

The only requirement of a Power 795 system to support Active Memory Mirroring is that in each node at least one processor module must be fully configured with eight dual inline memory modules (DIMMs).

Figure 3-3 shows the layout of a processor book and its components.

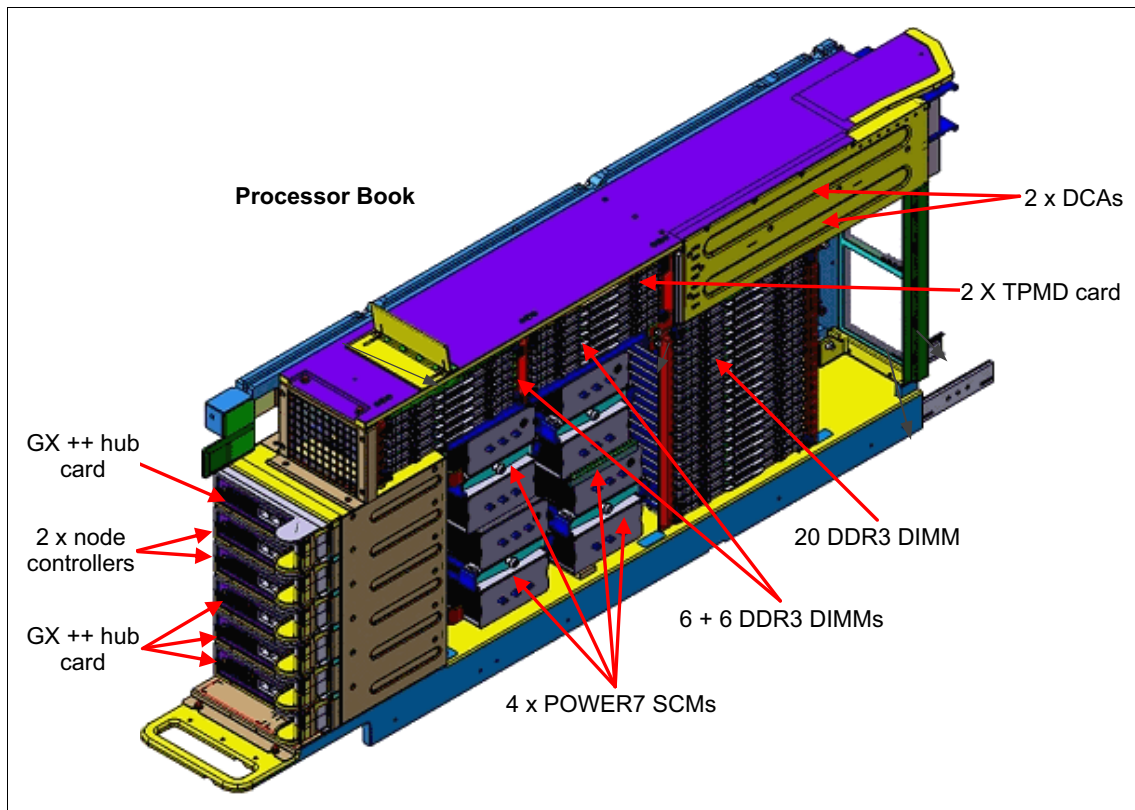


Figure 3-3 A POWER7 processor book and its components

Besides the hypervisor itself, other components that are vital to the server operation are also mirrored:

- ▶ Hardware page tables (HPTs), responsible for tracking the state of the memory pages assigned to partitions
- ▶ Translation control entities (TCEs), responsible for providing I/O buffers for the partition's communications
- ▶ Memory used by the hypervisor to maintain partition configuration, I/O states, Virtual I/O information, and partition state

There are components that are not mirrored after they are not vital to the regular server operations and would require a larger amount of memory to accommodate its data:

- ▶ Advanced Memory Sharing Pool
- ▶ Memory used to hold the contents of platform dumps

Note: Active Memory Mirroring will *not* mirror partition data. It was designed to mirror only the hypervisor code and its components, allowing this data to be protected against a DIMM failure.

3.5 Memory virtualization technologies comparison

Table 3-1 shows a comparison of the main characteristics of the memory virtualization discussed on this chapter.

Table 3-1 Memory virtualization comparison

Feature	Active Memory Sharing	Active Memory Expansion ^a	Active Memory Deduplication ^b	Active Memory Mirroring ^c
Operating system support	AIX, IBM i and Linux	AIX	AIX, IBM i and Linux	AIX, IBM i and Linux ^d
Licensing	PowerVM Enterprise Edition licensed per active processor	Feature code #4791 or feature code #4792 licensed per server	PowerVM Enterprise Edition licensed per active processor	Enabled on Power 780 and Power 795. Requires feature code #4797 for Power 770
I/O adapters	Only virtual I/O adapters supported	Virtual and physical I/O adapters supported	Only virtual I/O adapters supported	Virtual and physical I/O adapters supported
Processors	Only shared processor partitions supported	Shared processor partitions and dedicated processor partitions supported	Only shared processor partitions supported	Shared processor partitions and dedicated processor partitions supported
Configuration effort	Configuration on Virtual I/O Server and client partition level	Simple configuration on client partition level	Simple configuration on client partition level	Automatically enabled on supported hardware

a. Not a PowerVM feature. Available on POWER7 hardware and AIX technology.

b. PowerVM feature and works with partition configured with shared memory.

c. Not a PowerVM feature. Available on POWER7 high-end hardware

d. Operating System independent.

Note: Active Memory Sharing and Active Memory Expansion can be used in combination. Because of the higher complexity of such configuration, efforts for managing and problem determination can increase.



I/O virtualization overview

This chapter gives you an overview of I/O Virtualization on Power VM. It is organized into the following sections.

- ▶ Virtual I/O Server overview
- ▶ Storage virtualization overview
- ▶ Network virtualization overview

Combined with features designed into the POWER processors, the POWER Hypervisor delivers functions that enable other system technologies, including logical partitioning technology, virtualized processors, IEEE VLAN compatible virtual switch, virtual SCSI adapters, virtual Fibre Channel adapters and virtual consoles. The POWER Hypervisor is a basic component of the system's firmware and offers the following functions:

- ▶ Provides an abstraction between the physical hardware resources and the logical partitions that use them
- ▶ Enforces partition integrity by providing a security layer between logical partitions
- ▶ Controls the dispatch of virtual processors to physical processors.
- ▶ Saves and restores all processor state information during a logical processor context switch
- ▶ Controls hardware I/O interrupt management facilities for logical partitions

- ▶ Provides virtual LAN channels between logical partitions that help to reduce the need for physical Ethernet adapters for inter-partition communication
- ▶ Monitors the Service Processor and will perform a reset/reload if it detects the loss of the Service Processor, notifying the operating system if the problem is not corrected

The POWER Hypervisor is always active, regardless of the system configuration and also when not connected to the HMC. The POWER Hypervisor provides the following types of virtual I/O adapters:

- ▶ Virtual SCSI
- ▶ Virtual Ethernet
- ▶ Virtual Fibre Channel
- ▶ Virtual console

4.1 Virtual I/O Server overview

As part of PowerVM, the Virtual I/O Server is a software appliance with which you can associate physical resources and that allows you to share these resources among multiple client logical partitions. The Virtual I/O Server can provide both virtualized storage and network adapters, making use of the virtual SCSI and virtual Ethernet facilities.

For storage virtualization, these backing devices can be used:

- ▶ Direct-attached entire disks from the Virtual I/O Server
- ▶ SAN disks attached to the Virtual I/O Server
- ▶ Logical volumes defined on either of the aforementioned disks
- ▶ File-backed storage, with the files residing on either of the aforementioned disks
- ▶ Logical units from shared storage pools
- ▶ Optical storage devices.
- ▶ Tape storage devices

For virtual Ethernet, we can define Shared Ethernet Adapters on the Virtual I/O Server, bridging network traffic between the server internal virtual Ethernet networks and external physical Ethernet networks.

The Virtual I/O Server technology facilitates the consolidation of LAN and disk I/O resources and minimizes the number of physical adapters that are required, while meeting the non-functional requirements of the server.

The Virtual I/O Server can run in either a dedicated processor partition or a micro-partition.

The Virtual I/O Server also provides functionality for features such as Active Memory Sharing or Suspend/Resume and is a prerequisite if you want to use IBM Systems Director VMControl.

Figure 4-1 shows a very basic Virtual I/O Server configuration. This diagram only shows a small subset of the capabilities to illustrate the basic concept of how the Virtual I/O Server works. As you can see, the physical resources such as the physical Ethernet adapter and the physical disk adapter are accessed by the client partition using virtual I/O devices.

Figure 4-1 illustrates such a basic Virtual I/O Server configuration.

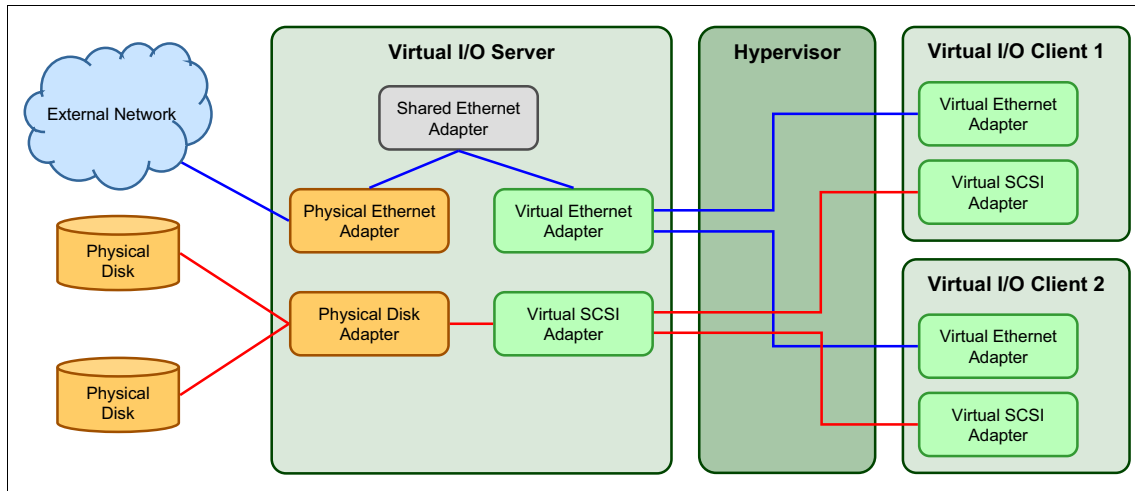


Figure 4-1 Simple Virtual I/O Server configuration

4.1.1 Supported platforms

The Virtual I/O Server can run on any POWER5 or later server which has the PowerVM Standard feature enabled. Also supported are IBM BladeCenter® Power Blade servers. With the PowerVM Standard Edition or the PowerVM Enterprise Edition Virtual I/O Servers can be deployed in pairs to provide high availability.

To understand the Virtual I/O Server support for physical network and storage devices, see the following website:

<http://www14.software.ibm.com/webapp/set2/sas/f/vios/documentation/data sheet.html>

4.1.2 Virtual storage mapping

Virtual I/O Server allows virtualization of physical storage resources. Virtualized storage devices are accessed by client partitions by one of these methods:

Virtual SCSI

Provides standard SCSI compliant access by client partitions to disk devices, optical devices and tape devices.

Virtual Fibre Channel devices

Provides access by Virtual Fibre Channel (VFC) to Fibre Channel attached disk and tape libraries.

Virtualized storage devices can be backed by the following types of physical storage devices:

Internal physical disks	Server internal disks such as SCSI, SAS or SATA attached disks located in I/O drawers.
External LUNs	LUNs residing on external storage subsystems accessed through Fibre Channel, or Fibre Channel over Ethernet, from IBM as well as certain third-party storage manufacturers. See “External storage subsystems” on page 41 for more detailed information about supported solutions.
Optical devices	Devices such as DVD-RAM, DVD-ROM and CD-ROM. Writing to a shared optical device is currently limited to DVD-RAM. DVD+RW and DVD-RW are not supported. A virtual optical device can only be assigned to one client partition at a time.
Tape devices	Devices such as SAS or USB attached tape devices. A virtual tape device can only be assigned to one client partition at a time.

Additionally, the following logical storage devices can be used to back virtualized storage devices:

Logical volumes	Internal disks as well as LUNs residing on external storage subsystems can be split into logical volumes on the Virtual I/O Server and then be exported to the client partitions.
Logical volume storage pools	A logical volume storage pool is a collection of internal or external disks split up into logical volumes that are used as backing devices for virtualized storage devices.
File storage pools	File storage pools are always part of a logical volume storage pool. A file storage pool contains files that are used as backing devices for virtualized storage devices.
Shared storage pools	Shared storage pools provide distributed access to storage resources using a cluster. Shared storage pools use files called logical units as backup devices for virtualized storage devices.
Virtual media repository	The virtual media repository provides a container for file backed optical media files such as ISO images. Only one virtual media repository is available per Virtual I/O Server

Figure 4-2 illustrates the aforementioned concepts.

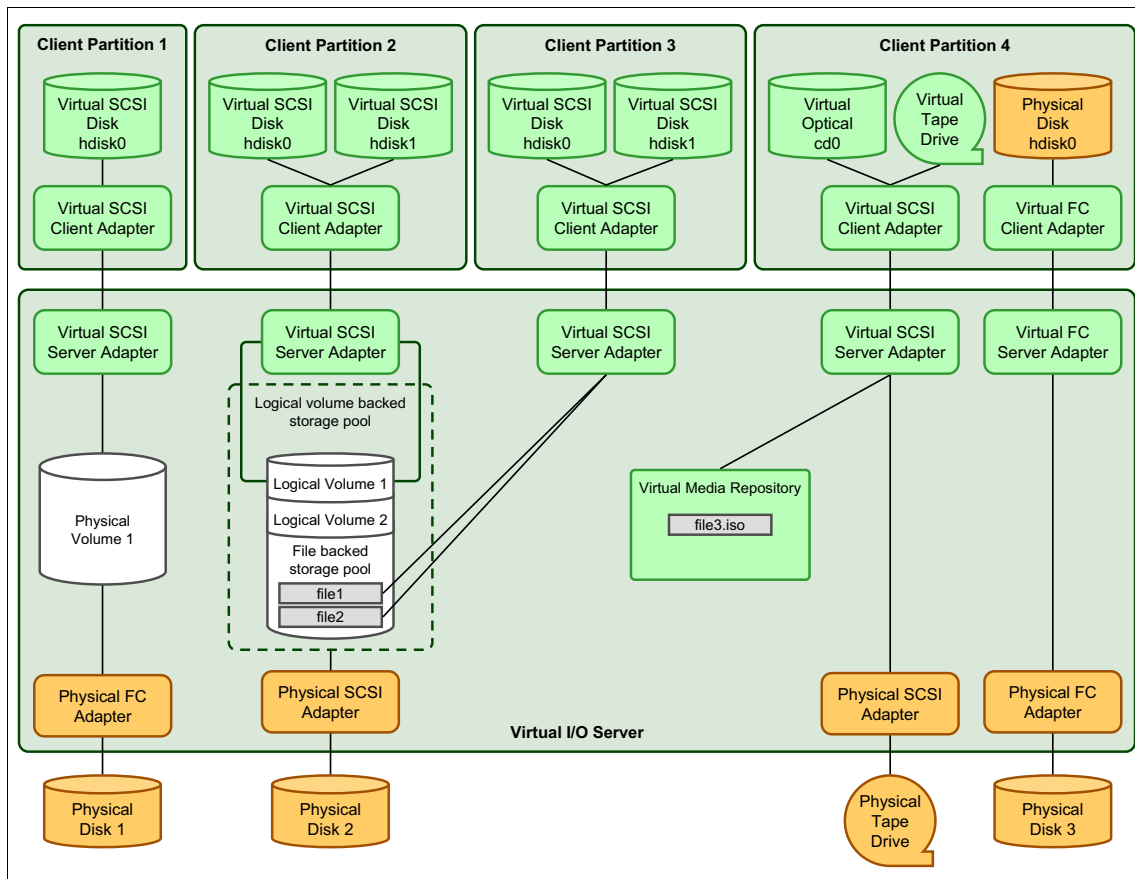


Figure 4-2 Virtual I/O Server concepts

Client partition 1 on the left side of the diagram has a virtual SCSI disk assigned, which is backed by a whole physical volume accessed through a Fibre Channel adapter on the Virtual I/O Server.

Client partition 2 has two virtual SCSI disks assigned. On the Virtual I/O Server these disks are backed by two logical volumes that are part of a logical volume backed storage pool. The logical volume backed storage pool consists of a local physical disk that has been partitioned into several logical volumes.

Client partition 3 has two virtual SCSI disks assigned. On the Virtual I/O Server these disks are backed by two files that are part of the file backed storage pool. File backed storage pools are always part of a logical volume backed storage pool. A file backed storage pool is a logical volume inside a logical volume backed storage pool.

Although each of these three partitions has different backing devices, they appear in the same way as virtual SCSI disks in the client partitions.

Client partition 4 has a virtual optical device and virtual tape drive assigned. The virtual optical device is backed by an ISO image file that has been loaded into the virtual media repository. The virtual tape drive is backed by a physical tape drive on the Virtual I/O Server.

Additionally, client partition 4 has a whole physical disk assigned that is passed through by the Virtual I/O Server using Virtual Fibre Channel. In contrast to client partition 1, which has a whole physical disk assigned through virtual SCSI, the disk does not appear as a virtual SCSI device. It appears in the same way as if it was provided through a physical Fibre Channel adapter (for example, as an IBM MPIO FC 2107 device in case of an AIX client partition).

Shared storage pools: Be aware that Figure 4-2 on page 40 does not show a shared storage pool configuration. See 4.2.7, “Shared storage pools” on page 55 for an example configuration of a shared storage pool.

For more details on each of the above types of storage, see the respective sections in the planning part of this book.

External storage subsystems

A large number of IBM storage solutions are supported, including these:

- ▶ IBM XIV® Storage System
- ▶ IBM DS8000® series
- ▶ IBM DS6000™ series
- ▶ IBM DS4000® series
- ▶ NSeries network attaches storage with Fibre Channel or iSCSI attach
- ▶ IBM Enterprise Storage Server® (ESS)
- ▶ SAN Volume Controller (SVC)

Important: IBM does not support virtualization of iSCSI LUNs using SW initiator. That is, an hdisk that is created on the VIOS as a result of iSCSI SW initiator attachment cannot be mapped to a client as a vSCSI disk on the client. However, we do continue to support virtualizing iSCSI LUNs using the TCP/IP Offload Engine (TOE) adapter (withdrawn, but might be available in some enterprises) with IBM System Storage N series attached through the TOE adapter.

An alternative recommendation is setting up a Shared Ethernet Adapter (SEA) on the VIOS and then running the iSCSI SW initiator on the client instead of in the VIOS.

Additionally, the Virtual I/O Server has been tested on selected configurations using third-party storage subsystems. You can find a list of supported configurations on the Virtual I/O Server support website or on the IBM System Storage® Interoperation Center (SSIC) website.

For the Virtual I/O Server support website, see the following website:

<http://www14.software.ibm.com/webapp/set2/sas/f/vios/documentation/data-sheet.html>

For the System Storage Interoperation Center (SSIC), see the following website:

<http://www-03.ibm.com/systems/support/storage/config/ssic>

Peer to Peer Remote Copy

Peer to Peer Remote Copy (PPRC) is a block level data replication mechanism available on IBM System Storage disk subsystem products, and is the underlying technology used by the Metro Mirror and Global Mirror features as defined here:

Metro Mirror	A mechanism whereby a disk mirroring relationship is established between a primary (source) volume and a secondary (target) volume such that both volumes are updated simultaneously. It is based on synchronous PPRC.
Global Mirror	A mechanism to provide data replication over extended distances between two sites for disaster recovery and business continuity. It is based on asynchronous implementation of PPRC.

When using these configurations, the storage system typically provides limited (read-only) access to the PPRC target device to avoid data corruption. Prior to Virtual I/O Server 2.2, configuring PPRC target devices on a Virtual I/O Server produced mixed results due to the ways various storage subsystems respond when a PPRC target is accessed.

The **mkvdev** command generally requires a disk device to be able to be opened in order to create virtual target devices. Virtual I/O Server 2.2 and newer provides an attributed called “mirrored” to the **mkvdev** command, which enables the system administrator to explicitly identify PPRC target devices when creating a virtual disk mapping. When this flag is used, the Virtual I/O Server uses an alternative method to access the disk, which allows it to successfully create the virtual target device.

This allows the virtual client partitions to access the PPRC target, although access will still be restricted to the limitations imposed by the storage system. When the PPRC relationship is removed or reversed, the client partition will gain read/write access to the device.

With the mirrored parameter, the system administrator can pre-configure the entire end-to-end client configuration. This saves time and reduces human error compared to attempting to configure the mappings during a fail-over event.

PPRC: There is no standard mechanism in the various storage systems to detect and report that a given disk belongs to a PPRC pair and that it is functioning as a PPRC primary or secondary. Hence the mirrored attribute depends upon the system administrator to identify PPRC targets at the time the virtual target device is created.

4.1.3 Virtual I/O Server network security

The Virtual I/O Server supports OpenSSH for secure remote logins. It also provides a firewall for limiting access by ports, network services, and IP addresses.

Starting with Virtual I/O Server 1.5 or later, an expansion pack is provided that delivers additional security functions, including these:

SNMP v3	SNMPv3 provides secure access by a combination of authenticating and encrypting packets over the network.
Kerberos	Kerberos is a system that provides a central authentication mechanism for a variety of client/server applications using passwords and secret keys.
LDAP	LDAP is a directory service that can be used for centralized user management.

4.1.4 Command line and cfgassist interface

The Virtual I/O Server provides a command line interface to perform management tasks such as these:

- ▶ Management of mappings between physical and virtual resources
- ▶ Gathering and displaying utilization data of resources through commands such as **topas**, **vmstat**, **iostat**, or **viostat** for performance management
- ▶ Troubleshooting of physical and virtual resources using the hardware error log
- ▶ Updating the Virtual I/O Server
- ▶ Securing the Virtual I/O Server by configuring user security and firewall policies

The Virtual I/O Server also provides a SMIT style menus using the **cfgassist** command. and through this most of the tasks can be performed.

4.1.5 Hardware Management Console integration

The Hardware Management Console (HMC) provides functions to simplify the handling of the Virtual I/O Server environment. For example, an overview of the Virtual Ethernet and Virtual SCSI topologies is available. It is also possible to execute Virtual I/O Server commands from the HMC.

4.1.6 System Planning Tool support

Using the System Planning Tool, complex Virtual I/O Server configurations such as these can be planned and deployed:

- ▶ Virtual SCSI adapter configuration
- ▶ Virtual Fibre Channel configuration
- ▶ Virtual Ethernet adapter configuration
- ▶ Shared Ethernet Adapter configuration with failover and EtherChannel

More information about how to use the System Planning Tool can be found in Appendix H, “System Planning Tool” on page 713.

4.1.7 Integrated Virtualization Manager

The Integrated Virtualization Manager (IVM) is used to manage selected Power Systems servers using a Web-based graphical interface without requiring an HMC.

This reduces the hardware needed for the adoption of virtualization technology, particularly for low-end systems. This solution fits in small and functionally simple environments where only few servers are deployed or some advanced HMC-like functions are required.

The Integrated Virtualization Manager (IVM) is a basic hardware management solution, included in the VIO software that inherits key Hardware Management Console (HMC) features.

4.1.8 Tivoli support

Included with the Virtual I/O Server are a number of pre-installed IBM Tivoli® agents that allow easy integration into an existing Tivoli Systems Management infrastructure.

Tivoli Storage Manager (TSM) client

The Tivoli Storage Manager (TSM) client can be used to back up Virtual I/O Server configuration data to a TSM server.

More information about TSM can be found at this website:

<http://www-306.ibm.com/software/tivoli/products/storage-mgr/>

IBM Tivoli Application Dependency Discovery Manager

IBM Tivoli Application Dependency Discovery Manager (TADDM) provides deep-dive discovery for Power Systems including their dependencies on the network and applications along with its configuration data, subsystems, and virtualized LPARs. TADDM is currently capable of recognizing a Virtual I/O Server and the software level it is running.

More information about TADDM can be found at this website:

<http://www-306.ibm.com/software/tivoli/products/taddm/>

IBM Tivoli Usage and Accounting Management agent

The IBM Tivoli Usage and Accounting Management (ITUAM) agent can be used to collect accounting and usage data of Virtual I/O Server resources so that they can be fed into ITUAM where they can be analyzed and used for reporting and billing.

More information about ITUAM can be found at this website:

<http://www-306.ibm.com/software/tivoli/products/usage-accounting/>

Tivoli Identity Manager

Tivoli Identity Manager (TIM) provides a secure, automated and policy-based user management solution that can be used to manage Virtual I/O Server users.

More information about TIM can be found at this website:

<http://www-306.ibm.com/software/tivoli/products/identity-mgr/>

IBM TotalStorage Productivity Center

Starting with Virtual I/O Server 1.5.2, you can configure the IBM TotalStorage Productivity Center agents on the Virtual I/O Server. TotalStorage Productivity Center is an integrated, storage infrastructure management suite that is designed to help simplify and automate the management of storage devices, storage networks, and capacity utilization of file systems and databases.

When you install and configure the TotalStorage Productivity Center agents on the Virtual I/O Server, you can use the TotalStorage Productivity Center user interface to collect and view information about the Virtual I/O Server.

You can then perform the following tasks using the TotalStorage Productivity Center user interface:

- ▶ Run a discovery job for the agents on the Virtual I/O Server.
- ▶ Run probes, run scans, and ping jobs to collect storage information about the Virtual I/O Server.
- ▶ Generate reports using the Fabric Manager and the Data Manager to view the storage information gathered.
- ▶ View the storage information gathered using the topology Viewer.

IBM Tivoli Monitoring

The Virtual I/O Server includes the IBM Tivoli Monitoring agent. Preinstalled are the IBM Tivoli Monitoring Premium Agent for VIOS (product code va) and the IBM Tivoli Monitoring CEC Agent. The IBM Tivoli Monitoring agent enables integration of the Virtual I/O Server into the IBM Tivoli Monitoring infrastructure and allows the monitoring of the health and availability of a Virtual I/O Server using the IBM Tivoli Enterprise Portal.

More information about IBM Tivoli Monitoring can be found at this website:

<http://www-01.ibm.com/software/tivoli/products/monitor>

IBM Tivoli Security Compliance Manager

Protects business against vulnerable software configurations in small, medium and large businesses by defining consistent security policies and monitor compliance of these defined security policies.

More information about IBM Tivoli Security Compliance Manager can be found at this website:

<http://www-01.ibm.com/software/tivoli/products/security-compliance-mgr>

4.1.9 Allowed third party applications

There are a number of third party applications that are allowed to be installed on the Virtual I/O Server. You can get a list of the allowed applications at this website:

http://www.ibm.com/partnerworld/gsd/searchprofile.do?name=VIOS_Recognized_List

Although these applications are allowed to be installed, IBM does not provide support for them. In case of problems, contact the application vendor.

4.1.10 Performance Toolbox support

Included with the Virtual I/O Server is a Performance Toolbox (PTX) agent that extracts performance data. This data can be viewed through an X Windows GUI if you have licensed the AIX Performance Toolbox.

4.1.11 Virtual I/O Server Performance advisor

The Virtual I/O Server Performance advisor is an application that runs within the customer's Virtual I/O Server for a user specified amount of time (hours), which polls and collects key performance metrics before analyzing results and providing a health check report and proposes changes to the environment or areas to investigate further.

The goal of the Virtual I/O Server Performance advisor is not to provide another monitoring tool, but instead have an expert system view performance metrics already available to the customer and make assessments and recommendations based on the expertise and experience available within the IBM systems performance group.

For more information on Virtual I/O Server Performance Advisor, see *IBM PowerVM Virtualization Managing and Monitoring*, SG24-7590.

4.2 Storage virtualization overview

Provisioning storage to servers can be done in the below two most popular ways

- ▶ Integrated disks:

With integrated server disks growing ever larger, requiring fewer disks for a given amount of storage, a significant cost can be associated with the adapters and the attachment of these disks to servers. With such large disks, it is also more difficult to utilize all the available space.

- ▶ External storage subsystems, for example, SAN disks or NAS disks:

Again with the introduction of ever larger and cheaper disks driving down the costs per gigabyte of storage, this leaves the costs of adapters (and additionally any switches and cabling) as a significant investment if a number of servers are involved.

In many cases it is beneficial to combine the storage requirements through a single adapter to better utilize the available bandwidth, taking into account the cost savings of not only the server related cost such as adapters or I/O slots, but also these components:

- ▶ Switches and switch ports (SAN or Etherchannel)
- ▶ Purchase of cables
- ▶ Installation of cables and patching panels

Very quickly the cost benefits for virtualizing storage can be realized, and that is before considering any additional benefits from the simplification of processes or organization in the business.

PowerVM on the IBM Power Systems platform supports up to 10 partitions per processor, up to a maximum of 254 partitions per server. With each partition typically requiring one I/O slot for disk attachment and a second Ethernet attachment, at least 508 I/O slots are required when using dedicated physical adapters, and that is before any resilience or adapter redundancy is considered.

Whereas the high-end IBM Power Systems can provide such a high number of physical I/O slots by attaching expansion drawers, the mid-end systems typically have a lower maximum number of I/O ports.

To overcome these physical requirements, I/O resources can be shared. Virtual SCSI and virtual Fibre Channel, provided by the Virtual I/O Server, provide the means to do this.

Most customers are deploying a pair of Virtual I/O Servers per physical server and using multipathing or mirroring technology to provide resilient access to storage. This configuration provides continuous availability to the disk resources, even if there is a requirement to perform maintenance on the Virtual I/O Servers.

Terms: You will see different terms in this book that refer to the various components involved with virtual SCSI. Depending on the context, these terms might vary. With SCSI, usually the terms *initiator* and *target* are used, so you might see terms such as *virtual SCSI initiator* and *virtual SCSI target*. On the HMC or IVM, the terms *virtual SCSI server adapter* and *virtual SCSI client adapter* are used to refer to the initiator and target, respectively.

4.2.1 Virtual SCSI

Virtual SCSI is used to refer to a virtualized implementation of the SCSI protocol. Virtual SCSI requires POWER5 or later hardware with the PowerVM feature activated. It provides virtual SCSI support for AIX, IBM i (requires POWER6 or later), and supported versions of Linux.

The following sections describe the virtual SCSI architecture.

Virtual SCSI client and server architecture overview

Virtual SCSI is based on a client/server relationship. The Virtual I/O Server owns the physical resources and acts as server or, in SCSI terms, target device. The client logical partitions access the virtual SCSI backing storage devices provided by the Virtual I/O Server as clients.

Interaction between client and server

The virtual I/O adapters are configured using an HMC or through the Integrated Virtualization Manager on smaller systems. The interaction between a Virtual I/O Server and an AIX, IBM i, or Linux client partition is enabled when both the *virtual SCSI server adapter* configured in the Virtual I/O Server's partition profile and the *virtual SCSI client adapter* configured in the client partition's profile have mapped slot numbers, and both the Virtual I/O Server and client operating system recognize their virtual adapter.

Dynamically added virtual SCSI adapters are recognized on the Virtual I/O Server after running the **cfgdev** command and on an AIX client partition after running the **cfgmgr** command. For IBM i and Linux, this additional step is not required; these operating systems will automatically recognize dynamically added virtual SCSI adapters.

After the interaction between virtual SCSI server and virtual SCSI client adapters is enabled, mapping storage resources from the Virtual I/O Server to the client partition is needed. The client partition configures and uses the storage resources when it starts up or when it is reconfigured at runtime.

The process runs as follows:

- ▶ The HMC maps interaction between virtual SCSI adapters.
- ▶ The mapping of storage resources is performed in the Virtual I/O Server.
- ▶ The client partition recognizes the newly mapped storage either dynamically as IBM i or Linux does, or after it has been told to scan for new devices, for example, after a reboot or by running the **cfgmgr** command on AIX.

4.2.2 Virtual Fibre Channel

N_Port ID Virtualization (NPIV) is an industry-standard technology that allows an NPIV capable Fibre Channel adapter to be configured with multiple virtual world-wide port names (WWPNs). This technology also is called Virtual Fibre Channel. Similar to the virtual SCSI functionality, virtual Fibre Channel is another way of securely sharing a physical Fibre Channel adapter among multiple Virtual I/O Server client partitions.

From an architectural perspective, the key difference with virtual Fibre Channel compared to virtual SCSI is that the Virtual I/O Server does not act as a SCSI emulator to its client partitions but as a direct Fibre Channel pass-through for the Fibre Channel Protocol I/O traffic through the POWER Hypervisor. Instead of generic SCSI devices presented to the client partitions with virtual SCSI, with virtual Fibre Channel, the client partitions are presented with native access to the physical SCSI target devices of SAN disk or tape storage systems.

The benefit with Virtual Fibre Channel is that the physical target device characteristics such as vendor or model information remain fully visible to the Virtual I/O Server client partition, so that device drivers such as multipathing software, middleware such as copy services, or storage management applications that rely on the physical device characteristics do not need to be changed.

Virtual Fibre Channel can be used for virtual disk and/or virtual tape. With NPIV, a Virtual Fibre Channel (VFC) server adapter in the Virtual I/O Server is mapped on the one hand to a port of the physical FC adapter, and on the other hand to a VFC client adapter from the client partition, as shown in the basic Virtual Fibre Channel configuration in Figure 4-3.

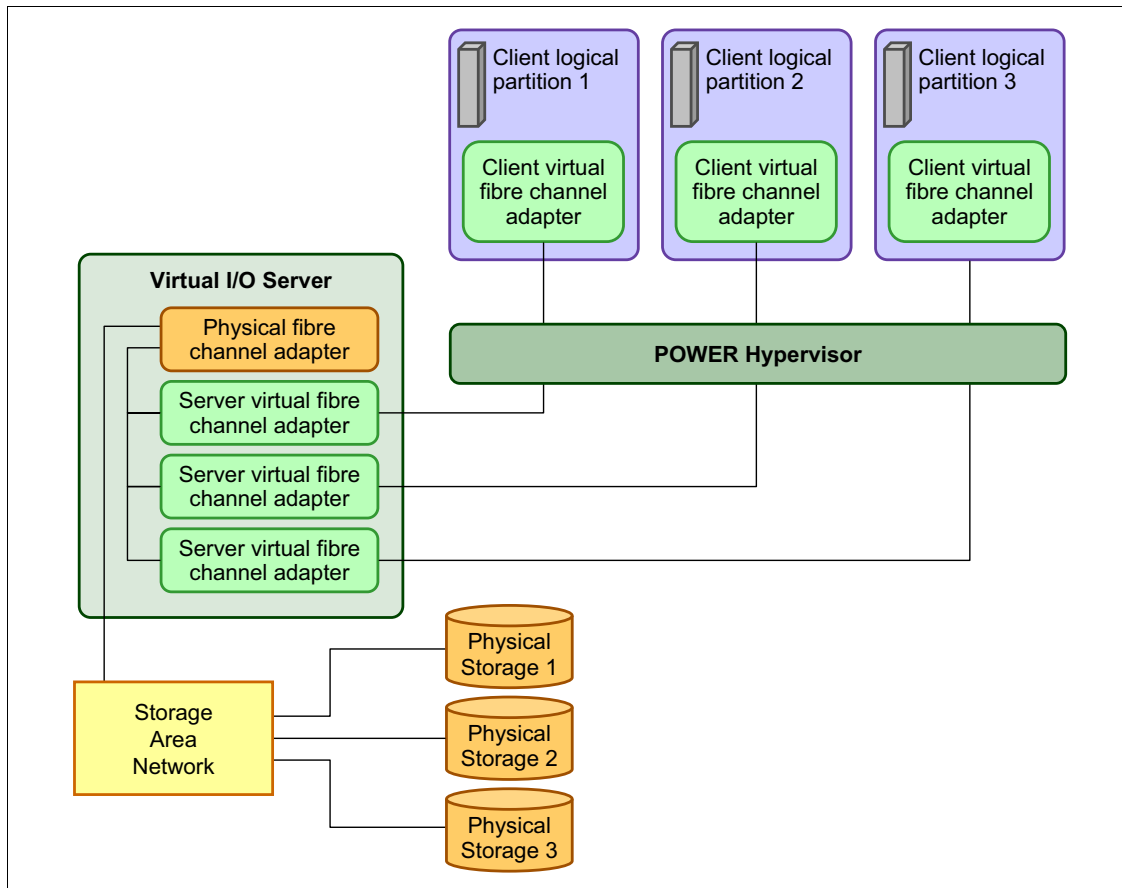


Figure 4-3 Virtual I/O Server virtual Fibre Channel adapter mappings

Two unique virtual world-wide port names (WWPNs) starting with the letter **c** are generated by the HMC for the VFC client adapter, which after activation of the client partition, log into the SAN like any other WWPNs from a physical port so that disk or tape storage target devices can be assigned to them as if they were physical FC ports.

Tip: Unless using PowerVM Live Partition Mobility or Suspend/Resume, only the first of the two created virtual WWPNs of a VFC client adapter is used.

The following considerations apply when using Virtual Fibre Channel:

- ▶ One VFC client adapter per physical port per client partition:
Intended to avoid a single point of failure
- ▶ Maximum of 64 active VFC client adapter per physical port:
Might be less due to other VIOS resource constraints
- ▶ Maximum of 64 targets per virtual Fibre Channel adapter
- ▶ 32,000 unique WWPN pairs per system platform:
 - Removing adapter does not reclaim WWPNs:
 - Can be manually reclaimed through CLI (mkyscfg, chhwres...)
 - Or use “virtual_fc_adapters” attribute

If these resources are exhausted, you can contact your IBM sales representative or Business Partner representative to purchase an activation code for more.

4.2.3 Virtual Optical

A DVD or CD device can be virtualized and assigned to Virtual I/O clients. Only one virtual I/O client can have access to the drive at a time. The advantage of a virtual optical device is that you do not have to move the parent SCSI adapter between virtual I/O clients.

Attention: The virtual optical drive cannot be moved to another Virtual I/O Server because client SCSI adapters cannot be created in a Virtual I/O Server. If you want the CD or DVD drive in another Virtual I/O Server, the virtual device must be de-configured and the parent SCSI adapter must be de-configured and moved, as described later in this section.

For more information, see 16.2.3, “Virtual optical” on page 490.

4.2.4 Virtual Tape

A tape device attached to a VIO Server can be virtualized and assigned to Virtual I/O clients. For more information, see 16.2.4, “Virtual tape” on page 492.

4.2.5 Virtual Console (virtual TTY/console support)

Each partition needs to have access to a system console. Tasks such as operating system install, network setup, and some problem analysis activities require a dedicated system console. For AIX and Linux, the POWER Hypervisor provides a virtual console using a virtual TTY or serial adapter and a set of POWER Hypervisor calls to operate on them.

Depending on the system configuration, the operating system console can be provided by the HMC or IVM virtual TTY.

For IBM i, an HMC managed server can use the 5250 system console emulation that is provided by a Hardware Management Console, or use an IBM i Access Operations Console. IVM managed servers must use an IBM i Access Operations Console.

Ports: The serial ports on an HMC-based system are inactive. Partitions requiring a TTY device must have an async adapter defined. The async adapter can be dynamically moved into or out of partitions with dynamic LPAR operations.

On the IVM, the serial ports are configured and active. They are used for initial configuration of the system.

4.2.6 Availability

Many customers want to set up highly available storage environments. Power VM allows them to configure the same using Virtual SCSI and Virtual Fibre Channel by having redundancy.

Redundancy of Virtual SCSI and Virtual Fibre Channel can be achieved using MPIO and LVM mirroring in the client partition and also in Virtual I/O Server. The following sections give you an overview of redundancy of Virtual SCSI and Virtual Fibre Channel in Power VM. You can find more details in the Planning and Setup parts of this book.

Redundancy of Virtual SCSI using Dual Virtual I/O Servers

Figure 4-4 shows one possible configuration in the Power VM environment that shows redundancy of Virtual SCSI using MPIO at client partitions. The diagram shows a Dual Virtual I/O Server environment where the client partition has two virtual SCSI client adapters and each of them is mapped to two different Virtual SCSI server adapters on different Virtual I/O Servers. Each Virtual I/O Server maps the same physical volume to the Virtual SCSI Server adapter on them.

The client partition sees the same hdisk, mapped from two Virtual I/O Servers using Virtual SCSI. To achieve this, the same storage needs to be zoned to Virtual I/O Servers from the storage subsystem. This configuration also has redundancy at the Virtual I/O Server physical Fibre Channel adapter.

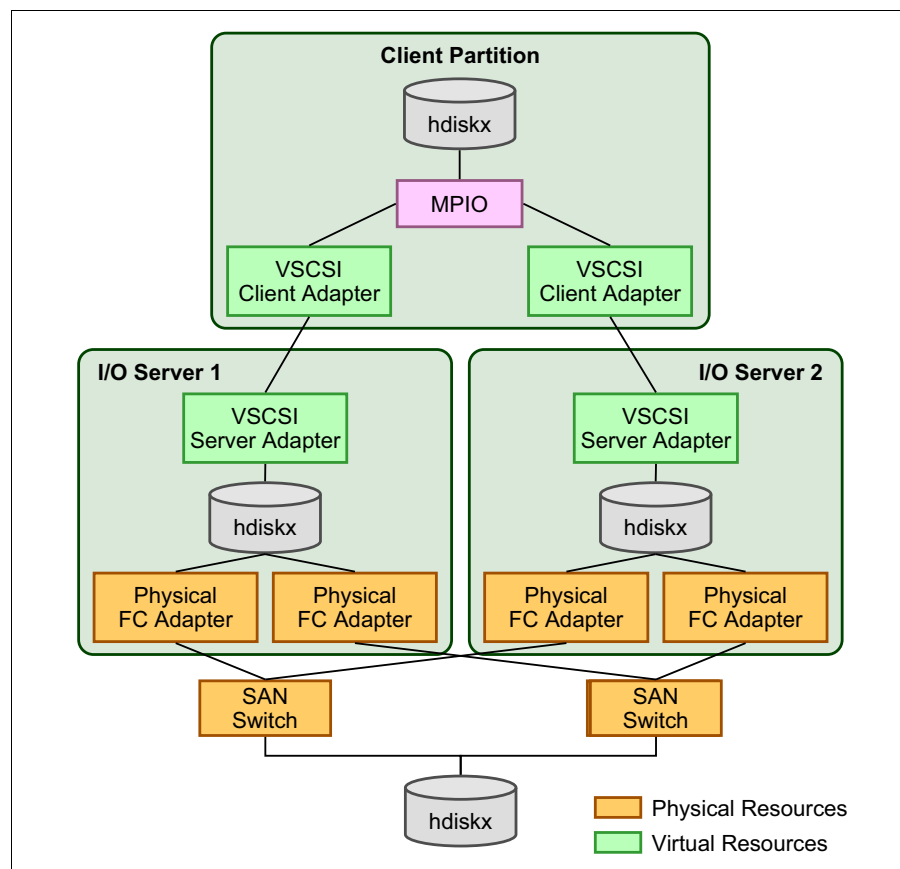


Figure 4-4 Redundancy of Virtual SCSI using Dual Virtual I/O Server

Redundancy of Virtual Fibre Channel

A host bus adapter and Virtual I/O Server redundancy configuration provides a more advanced level of redundancy for the virtual I/O client partition, as shown in Figure 4-5.

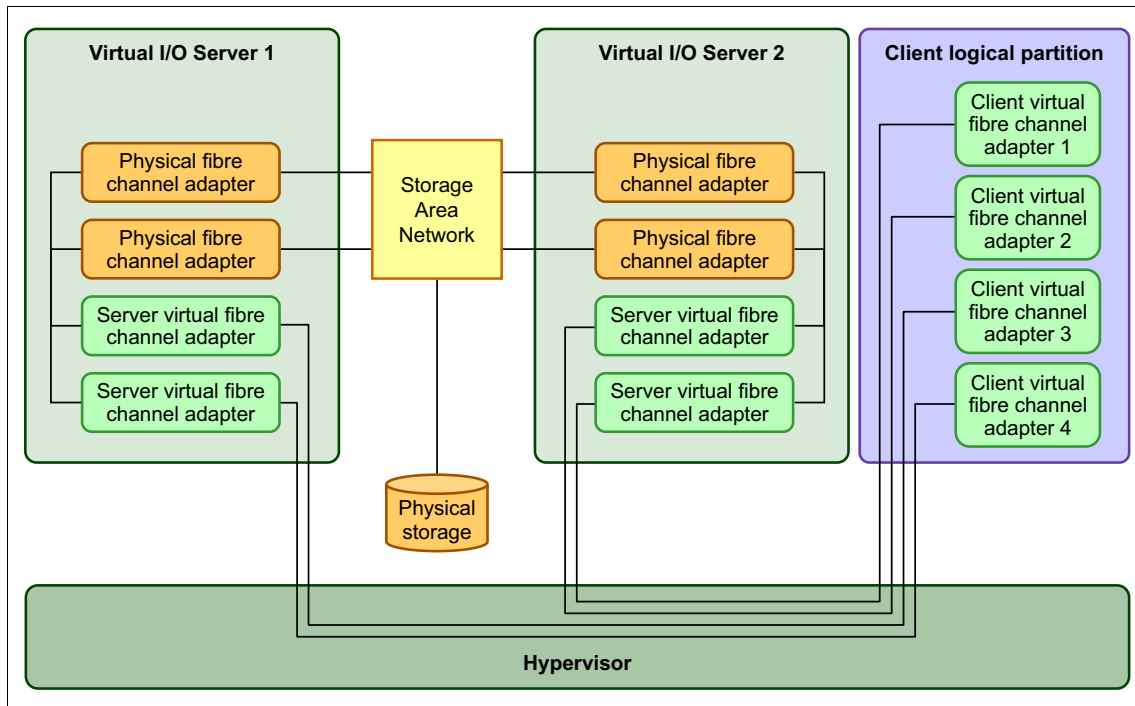


Figure 4-5 Host bus adapter and Virtual I/O Server failover

4.2.7 Shared storage pools

With Virtual I/O Server version 2.2.0.11 Fix Pack 11 Service Pack 1, shared storage pools are introduced. A shared storage pool is a server based storage virtualization that is clustered and is an extension of existing storage virtualization on the Virtual I/O Server.

Shared storage pools provide the following benefits:

- ▶ Simplify the aggregation of large numbers of disks across multiple Virtual I/O Servers.
- ▶ Improve the utilization of the available storage.
- ▶ Simplify administration tasks.

After the physical volumes are allocated to a Virtual I/O Server in the shared storage pool environment, the physical volume management tasks, such as a capacity management or an allocation of the volumes to a client partition, are performed by the Virtual I/O Server.

Shared Storage Pools also allow better utilization of the available storage by using thin provisioning. The thinly provisioned device is not fully backed by physical storage if the data block is not in actual use.

Architecture overview

A shared storage pool is a pool of SAN storage devices that can span multiple Virtual I/O Servers. It is based on a cluster of Virtual I/O Servers and a distributed data object repository with a global namespace. Each Virtual I/O Server that is part of a cluster represents a cluster node.

The distributed data object repository is using a cluster filesystem that has been developed specifically for the purpose of storage virtualization using the Virtual I/O Server. It provides redirect-on-write capability and is highly scalable. The distributed object repository is the foundation for advanced storage virtualization features, such as thin provisioning.

When using shared storage pools, the Virtual I/O Server provides storage through *logical units* that are assigned to client partitions. A logical unit is a file backed storage device that resides in the cluster filesystem in the shared storage pool. It appears as a virtual SCSI disk in the client partition, in the same way as a for example, a virtual SCSI device backed by a physical disk or a logical volume

Clustering model

The Virtual I/O Servers that are part of the shared storage pool are joined together to form a cluster. A Virtual I/O Server that is part of a cluster is also referred to as cluster node. Only Virtual I/O Server partitions can be part of a cluster.

The Virtual I/O Server clustering model is based on Cluster Aware AIX (CAA) and RSCT technology. The cluster for the shared storage pool is an RSCT Peer Domain cluster. Therefore a network connection is needed between all the Virtual I/O Servers that are part of the shared storage pool.

Figure 4-6 shows an abstract image of clustered Virtual I/O Servers.

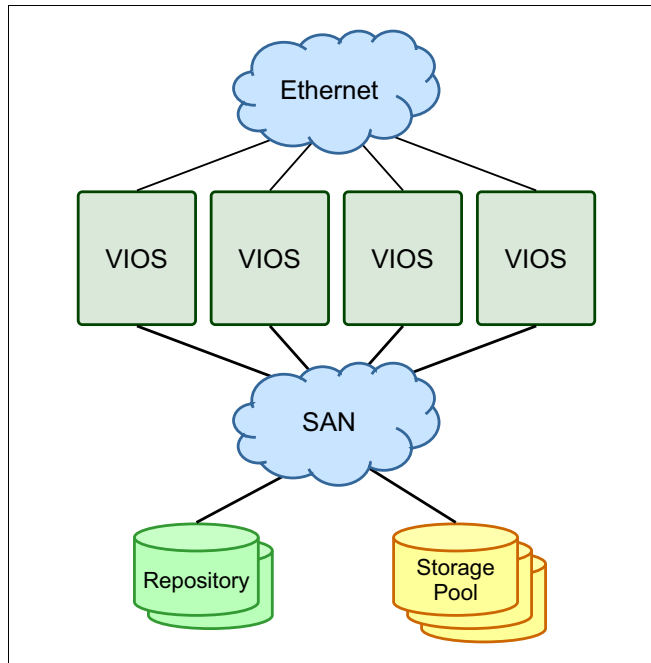


Figure 4-6 Abstract image of the clustered Virtual I/O Servers

On the Virtual I/O Server, the *pool*d daemon handles group services and is running in the user space. The *vio_daemon* daemon is responsible for monitoring the health of the cluster nodes and the pool as well as the pool capacity.

Each Virtual I/O Server in the cluster requires at least one physical volume for the repository that is used by the CAA subsystem and one or more physical volumes for the storage pool.

All cluster nodes in a cluster can see all the disks. Therefore the disks need to be zoned to all the cluster nodes that are part of the shared storage pools. All nodes can read and write to the shared storage pool. The cluster uses a distributed lock manager to manage access to the storage.

PowerVM model with Shared Storage Pools

Figure 4-7 shows an extension of a PowerVM model with shared storage pools.

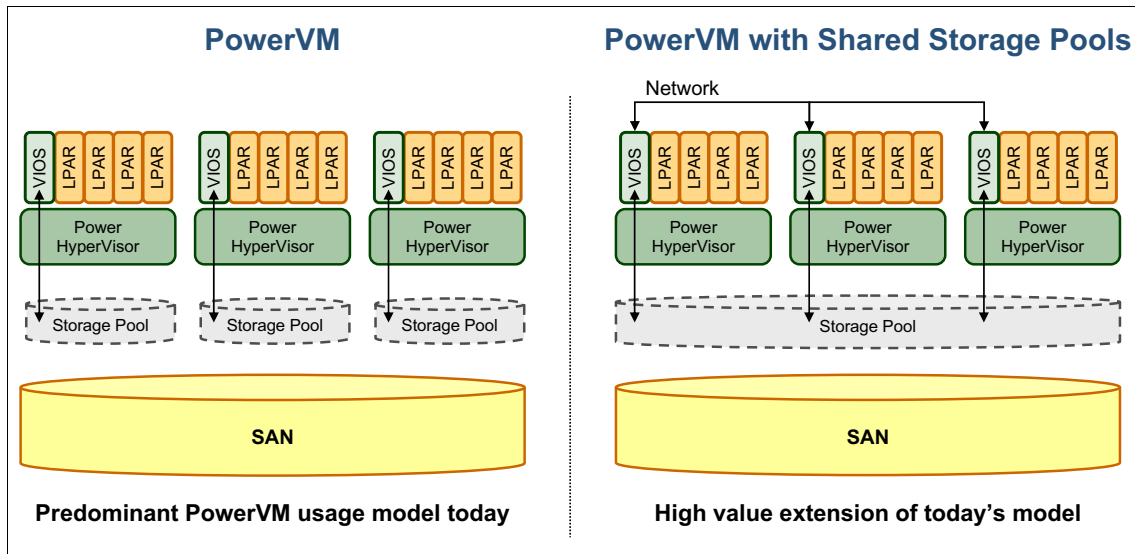


Figure 4-7 PowerVM Model with Shared Storage Pools

Notes:

On VIOS version 2.2.0.11, Fix Pack 24, Service Pack 1, you can create a cluster consists of only one VIOS partition, On VIOS version 2.2.1.3 or later, you can create a cluster that consists of 4 networked VIOS partitions. On VIOS Version 2.2.2.0 or later, a cluster can consists of up to 16 networked VIOS partitions.

Thin provisioning

A thin-provisioned device represents a larger image than the actual physical disk space it is using. It is not fully backed by physical storage as long as the blocks are not in use.

A thin-provisioned logical unit is defined with a user-specified size when it is created. It appears in the client partition as a virtual SCSI disk with that user-specified size. However, on a thin-provisioned logical unit, blocks on the physical disks in the shared storage pool are only allocated when they are used.

Compared to a traditional storage device, which allocates all the disk space when the device is created, this can result in significant savings in physical disk space. It also allows over-committing of the physical disk space.

Consider a shared storage pool that has a size of 20 GB. If you create a logical unit with a size of 15 GB, the client partition will see a virtual disk with a size of 15 GB. But as long as the client partition does not write to the disk, only a small portion of that space will initially be used from the shared storage pool. If you create a second logical unit also with a size of 15 GB, the client partition will see two virtual SCSI disks, each with a size of 15 GB. So although the shared storage pool has only 20 GB of physical disk space, the client partition sees 30 GB of disk space in total. After the client partition starts writing to the disks, physical blocks will be allocated in the shared storage pool and the amount of free space in the shared storage pool will decrease.

After the physical blocks are allocated to a logical unit to write actual data, the physical blocks allocated are not released from the logical unit until the logical unit is removed from the shared storage pool. Deleting files, file systems or logical volumes, which reside on the virtual disk from the shared storage pool, on a client partition does not increase free space of the shared storage pool.

When the shared storage pool is full, client partitions that are using virtual SCSI disks backed by logical units from the shared storage pool will see an I/O error on the virtual SCSI disk. Therefore even though the client partition will report free space to be available on a disk, that information might not be accurate if the shared storage pool is full.

To prevent such a situation, the shared storage pool provides a threshold that, if reached, writes an event in the errorlog of the Virtual I/O Server. The default threshold value is 75, which means an event is logged if the shared storage pool has less than 75% free space. The errorlog must be monitored for this event so that additional space can be added before the shared storage pool becomes full. The threshold can be changed using the **alert** command.

Example 4-1 shows a shared storage pool that initially has almost 40 GB of free space. The threshold is at the default value of 75. After the free space drops below 75%, the alert is triggered, as you can see from the **errlog** command output.

Example 4-1 Shared storage pool free space alert

```
$ alert -list -clustername clusterA -spname poolA
Pool Name                               PoolID
Threshold Percentage
poolA                                   15757390541369634258      75
$ lssp -clustername clusterA
Pool      Size(mb)  Free(mb)  LUs      Type    PoolID
poolA     40704     40142     1        CLPOOL  15757390541369634258
$ lssp -clustername clusterA
```

```

Pool                Size(mb)  Free(mb)  LUs      Type  PoolID
poolA               40704    29982    1        CLPOOL
15757390541369634258
$ errlog
IDENTIFIER  TIMESTAMP  T C RESOURCE_NAME  DESCRIPTION
0FD4CF1A    1214152010  I 0 VI01_-26893535  Informational Message

```

Figure 4-8 shows an image of two thin-provisioned logical units in a shared storage pool. As you can see, not all of the blocks of the virtual disk in the client partition are backed by physical blocks on the disk devices in the shared storage pool.

A logical unit cannot be resized after creation. If you need more space from the shared storage pool on the client partition, you can map an additional logical unit to the client partition or replace the existing logical unit with a larger one.

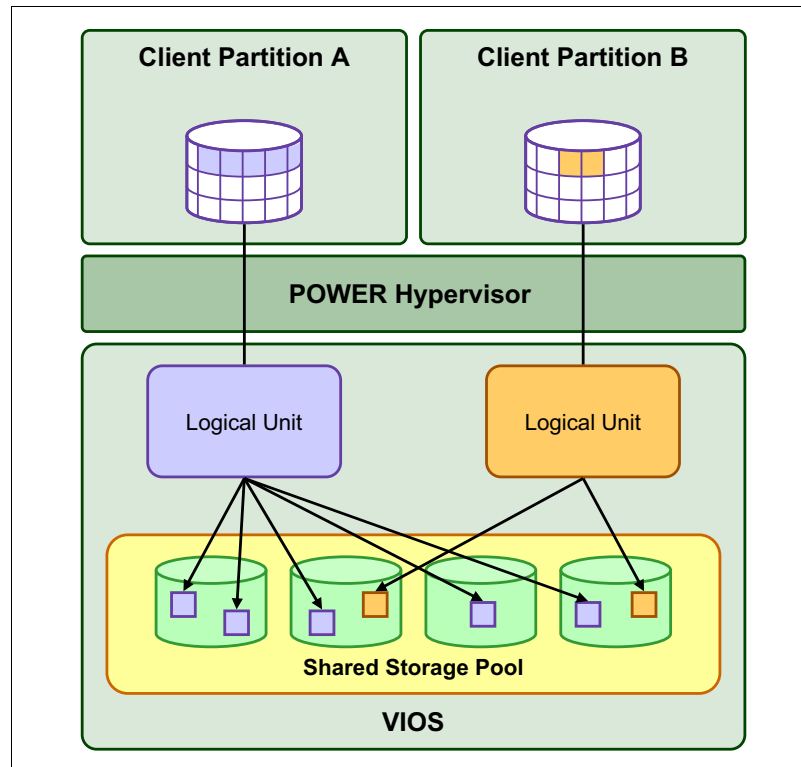


Figure 4-8 Thin-provisioned devices in the shared storage pool

The virtual target device for the logical unit from the shared storage pool is dependent on the following components:

- ▶ Shared storage pool
- ▶ Distributed object data repository
- ▶ Clustering infrastructure

If there is a problem with one of these components, the virtual target device will go into a state where it will fail most commands sent from the client partitions, including any I/O. If the dependent component recovers, the virtual target device also needs to recover.

Persistent reservation support

The virtual SCSI disk devices exported from the shared storage pool supports SCSI persistent reservations. These SCSI persistent reservations persist across hard resets, logical unit resets, or initiator target nexus loss. The persistent reservations supported by the virtual SCSI disk from the shared storage pool support the required features for the SCSI-3 Persistent Reserves standard.

The additional options, *PR_exclusive* and *PR_shared*, are added to the reserve policy for the virtual SCSI disk device from the shared storage pool. The *PR_exclusive* is a persistent reserve for exclusive hot access, and the *PR_shared* is a persistent reserve for shared hot access.

4.3 Network virtualization overview

POWER systems offer an extensive range of networking options. PowerVM enables further virtualization capabilities that can be used to provide greater flexibility and security, and to increase the utilization of the hardware.

It is easy to confuse virtual networking terms and technologies as many of them are named similarly. For clarity and reference, common terms are defined here:

Virtual Ethernet	Virtual Ethernet enables interpartition communication without the need for physical network adapters assigned to each partition.
Virtual Ethernet adapter	Virtual Ethernet adapters allow client logical partitions to send and receive network traffic without having a physical Ethernet adapter.
Virtual LAN	Virtual Local Area Networks (VLAN) allows the physical network to be logically segmented.
Virtual switches	An in-memory, hypervisor implementation of a Layer-2 switch.

Shared Ethernet Adapter A Virtual I/O Server software adapter that bridges physical and virtual Ethernet networks.

These concepts are described in more detail in the following sections.

4.3.1 Virtual Ethernet

Virtual Ethernet allows the administrator to define in-memory connections between partitions handled at the system level (POWER Hypervisor and operating systems interaction). These connections are represented as virtual Ethernet adapters and exhibit characteristics similar to physical high-bandwidth Ethernet adapters. They support the industry standard protocols, such as IPv4, IPv6, ICMP, or ARP.

Virtual Ethernet requires the following components:

- ▶ An IBM Power server (POWER5 or newer).
- ▶ The appropriate level of AIX (V5R3 or later), IBM i (5.4 or later) or Linux.
- ▶ Hardware Management Console (HMC) or Integrated Virtualization Manager (IVM) to define the virtual Ethernet adapters.

4.3.2 Virtual Ethernet adapter

Virtual Ethernet adapter is the component that will provide the communication to the LPAR transparently like with a physical device, but offering more flexibility. This is the base element to the further network configuration on the Virtual I/O Server (VIOS) and the client logical partitions (LPARs).

The virtual Ethernet adapter can be used:

- ▶ To configure an Ethernet interface with an IP address onto it
- ▶ To configure VLAN adapters (one per VID) onto it
- ▶ As a member of a Network Interface Backup adapter

But it cannot be used for EtherChannel or Link Aggregation.

Virtual Ethernet does not require the purchase of any additional features or software.

A highlight to the virtual Ethernet adapter is the ability to manage the Media Access Control (MAC) address. Despite of many operating systems are able to manage the MAC Address, the POWER system, through the virtual adapter, offers an easy manual management, which is transparent to the operating system and avoids additional configurations.

In a Power system, the hardware MAC address of a virtual Ethernet adapter is automatically generated by the HMC when it is defined. Enhancements introduced in POWER7 servers allow the partition administrator to do these tasks:

- ▶ Specify the hardware MAC address of the virtual Ethernet adapter at creation time.
- ▶ Restrict the range of addresses that are allowed to be configured by the operating system within the partition.

4.3.3 Virtual LAN

Virtual Local Area Network (VLAN) is a method to logically segment a physical network so that layer 2 connectivity is restricted to members that belong to the same VLAN. This separation is achieved by tagging Ethernet packets with their VLAN membership information and then restricting delivery to members of that VLAN. VLAN is described by the IEEE 802.1Q standard.

The VLAN tag information is referred to as VLAN ID (VID). Ports on a switch are configured as being members of a VLAN designated by the VID for that port. The default VID for a port is referred to as the Port VID (PVID). The VID can be added to an Ethernet packet either by a VLAN-aware host, or by the switch in the case of VLAN-unaware hosts. Ports on an Ethernet switch must therefore be configured with information indicating whether the host connected is VLAN-aware.

For VLAN-unaware hosts, a port is set up as untagged and the switch will tag all packets entering through that port with the Port VLAN ID (PVID). The switch will also untag all packets exiting that port before delivery to the VLAN unaware host. A port used to connect VLAN-unaware hosts is called an untagged port, and it can be a member of only one VLAN identified by its PVID.

Hosts that are VLAN-aware can insert and remove their own tags and can be members of more than one VLAN. These hosts are typically attached to ports that do not remove the tags before delivering the packets to the host, but will insert the PVID tag when an untagged packet enters the port.

A port will only allow packets that are untagged or tagged with the tag of one of the VLANs that the port belongs to. These VLAN rules are in addition to the regular Media Access Control (MAC) address-based forwarding rules followed by a switch. Therefore, a packet with a broadcast or multicast destination MAC is also delivered to member ports that belong to the VLAN that is identified by the tags in the packet. This mechanism ensures the logical separation of the physical network based on membership in a VLAN.

The use of VLAN technology provides more flexible network deployment over traditional network technology. It can help overcoming physical constraints of the environment and to reduce the number of required switches, ports, adapters, cabling, and uplinks. This simplification in physical deployment does not come for free: the configuration of switches and hosts becomes more complex when using VLANs. But the overall complexity is not increased; it is just shifted from physical to virtual.

4.3.4 Virtual switches

The POWER Hypervisor implements an IEEE 802.1Q VLAN style virtual Ethernet switch. Similar to a physical IEEE 802.1Q Ethernet switch, it can support tagged and untagged ports. A virtual switch does not really need ports, so the virtual ports correspond directly to virtual Ethernet adapters that can be assigned to partitions from the HMC or IVM. There is no need to explicitly attach a virtual Ethernet adapter to a virtual Ethernet switch port. To draw on the analogy of physical Ethernet switches, a virtual Ethernet switch port is configured when you configure the virtual Ethernet adapter on the HMC or IVM.

The POWER Hypervisor's virtual Ethernet switch can support virtual Ethernet frames of up to 65408 bytes size, which is much larger than what physical switches support: 1522 bytes is standard and 9000 bytes are supported with Gigabit Ethernet Jumbo Frames. Thus, with the POWER Hypervisor's virtual Ethernet, you can increase TCP/IP's MTU size to 65394 (= 65408 - 14 for the header, no CRC) in the non-VLAN case and to 65390 (= 65408 - 14 - 4 for the VLAN, again no CRC) if you use VLAN.

Increasing the MTU size can benefit performance because it may improve the efficiency of the transport. This depends on the communication data requirements of the running workload.

4.3.5 Shared Ethernet Adapter

A Shared Ethernet Adapter (SEA) is a Virtual I/O Server component that bridges a real Ethernet adapter and one or more virtual Ethernet adapters:

- ▶ The real adapter can be a physical Ethernet adapter, a Link Aggregation or EtherChannel device. The real adapter cannot be another Shared Ethernet Adapter or a VLAN pseudo-device.
- ▶ The virtual Ethernet adapter must be a virtual I/O Ethernet adapter. It cannot be any other type of device or adapter.

Using a Shared Ethernet Adapter, logical partitions on the virtual network can share access to the physical network and communicate with standalone servers and logical partitions on other systems. The Shared Ethernet Adapter eliminates the need for each client logical partition to a dedicated physical adapter to connect to the external network.

A Shared Ethernet Adapter provides access by connecting the internal VLANs with the VLANs on the external switches. Using this connection, logical partitions can share the IP subnet with standalone systems and other external logical partitions.

The Shared Ethernet Adapter forwards outbound packets received from a virtual Ethernet adapter to the external network and forwards inbound packets to the appropriate client logical partition over the virtual Ethernet link to that logical partition. The Shared Ethernet Adapter processes packets at layer 2, so the original MAC address and VLAN tags of the packet are visible to other systems on the physical network.

The Shared Ethernet Adapter has a bandwidth apportioning feature, also known as Virtual I/O Server quality of service (QoS). QoS allows the Virtual I/O Server to give a higher priority to some types of packets. In accordance with the IEEE 801.q specification, Virtual I/O Server administrators can instruct the Shared Ethernet Adapter to inspect bridged VLAN-tagged traffic for the VLAN priority field in the VLAN header. The 3-bit VLAN priority field allows each individual packet to be prioritized with a value from 0 to 7 to distinguish more important traffic from less important traffic. More important traffic is sent preferentially and uses more Virtual I/O Server bandwidth than less important traffic.

Shared Ethernet Adapters also support the following additional features:

Link Aggregation	Bundling of several physical network adapters into one logical device using EtherChannel functionality.
SEA failover	The SEA failover feature allows highly available configurations by using two Shared Ethernet Adapters running in two different Virtual I/O Servers.
TCP segmentation offload	The SEA supports the large send and large receive features.
GVRP	GVRP (GARP VLAN Registration Protocol) is a protocol that facilitates control of VLANs within larger networks. It helps to maintain VLAN configurations dynamically based on network adapter configurations.

4.4 Platform consideration for I/O virtualization

The main discussion in this publication is based on the AIX platform. Here we talk about the special considerations for the IBM i and Linux platforms.

4.4.1 I/O virtualization consideration for IBM i

IBM i, formerly known as IBM i5/OS™ and IBM OS/400®, has a long virtual I/O heritage. It has been able to be a host partition for other clients such as AIX (since i5/OS V5R3), Linux (since OS/400 V5R1), or Windows by using a virtual SCSI connection for network server storage spaces and using virtual Ethernet.

Starting with a more recent operating system level, IBM i 6.1, the IBM i virtualization support was extended on IBM POWER Systems POWER6 models or later. IBM i can now support being a client partition itself, to either the PowerVM Virtual I/O Server or another IBM i 6.1 or later hosting partition.

For a list of supported IBM System Storage storage systems with IBM i as a client of the Virtual I/O Server, see the *IBM i Virtualization and Open Storage Read-me First* at the following website:

http://www-03.ibm.com/systems/resources/systems_i_Virtualization_Open_Storage.pdf

The following sections describe the IBM i virtual I/O support as a client of the PowerVM Virtual I/O Server.

Virtual Ethernet

IBM i supports various virtual Ethernet implementations available on IBM Power Systems, as follows:

- ▶ *Virtual LAN (VLAN)* for inter-partition communication on the same IBM Power Systems server through a 1 Gb virtual Ethernet switch in the POWER hypervisor based on the IEEE 802.1Q VLAN standard.
- ▶ *Shared Ethernet Adapter (SEA)* virtual Ethernet (including SEA failover) provided by the PowerVM Virtual I/O Server acting an OSI layer 2 bridge between a physical Ethernet adapter and up to 16 virtual Ethernet adapters each supporting up to 20 VLANs.

Up to 32767 virtual Ethernet adapters are supported for each IBM i logical partition that can belong to a maximum of 4094 virtual LANs.

Support: IBM i does not support IEEE 802.1Q VLAN tagging.

To implement Ethernet redundancy for IBM i, one of the following methods can be used:

- ▶ Shared Ethernet Adapter (SEA) failover with two Virtual I/O Servers.
- ▶ Virtual IP Address (VIPA) failover with two or more physical Ethernet ports.

Considerations:

- ▶ Ethernet Link Aggregation requires corresponding port aggregation to be configured on the Ethernet switch and is not supported for virtual Ethernet ports connected to the virtual Ethernet switch of the POWER Hypervisor.
- ▶ Using VIPA failover with virtual Ethernet adapters as used in a SEA environment does not work without special customization such as a scripting solution for periodic link health checks, because physical link state changes are not propagated to the virtual adapter seen by the IBM i client.

From an IBM i client perspective, a virtual Ethernet adapter reports in as a model 268C and type 002 for the virtual IOP/IOA and port as shown in Figure 4-9. It needs to be configured with an Ethernet line description and interface just like a physical Ethernet adapter.

Logical Hardware Resources Associated with IOP			
Type options, press Enter.			
2=Change detail 4=Remove 5=Display detail 6=I/O debug			
7=Verify 8=Associated packaging resource(s)			
Opt Description	Type-Model	Status	Resource Name
Virtual IOP	268C-002	Operational	CMB06
Virtual Comm IOA	268C-002	Operational	LIN03
Virtual Comm Port	268C-002	Operational	CMN03
F3=Exit F5=Refresh F6=Print F8=Include non-reporting resources			
F9=Failed resources F10=Non-reporting resources			
F11=Display serial/part numbers F12=Cancel			

Figure 4-9 Virtual Ethernet adapter reported on IBM i

Virtual SCSI

From a disk storage perspective, IBM i 6.1 or later, as a client partition of the PowerVM Virtual I/O Server, offers completely new possibilities for IBM i external storage attachment. Instead of IBM i 520 bytes/sector formatted storage, which includes 8 bytes header information and 512 bytes data per sector, the Virtual I/O Server attaches to industry standard 512 bytes/sector formatted storage. This now allows common 512 bytes/sector storage systems such as the supported IBM midrange storage systems or the IBM SAN Volume Controller to be attached to IBM i by the Virtual I/O Server’s virtual SCSI interface.

To make IBM i compatible with 512 bytes/sector storage, the POWER hypervisor has been enhanced for POWER6 servers or later to support conversion from 8 x 520 bytes/sector pages into 9 x 512 bytes/sector pages.

The additional 9th sector, called an iSeries Specific Information (ISSI) sector, is used to store the 8 bytes header information from each of the 8 x 520 bytes sectors of a page so they fit into 9 x 512 bytes. To ensure data atomicity, that is, ensuring that all 9 sectors now representing a 4 KB page are processed as an atomic block, 8 bytes of control information are added so that in addition to the headers, also 64 bytes of user data are shifted into the 9th sector. Figure 4-10 illustrates the 520-bytes to 512-bytes sector page conversion.

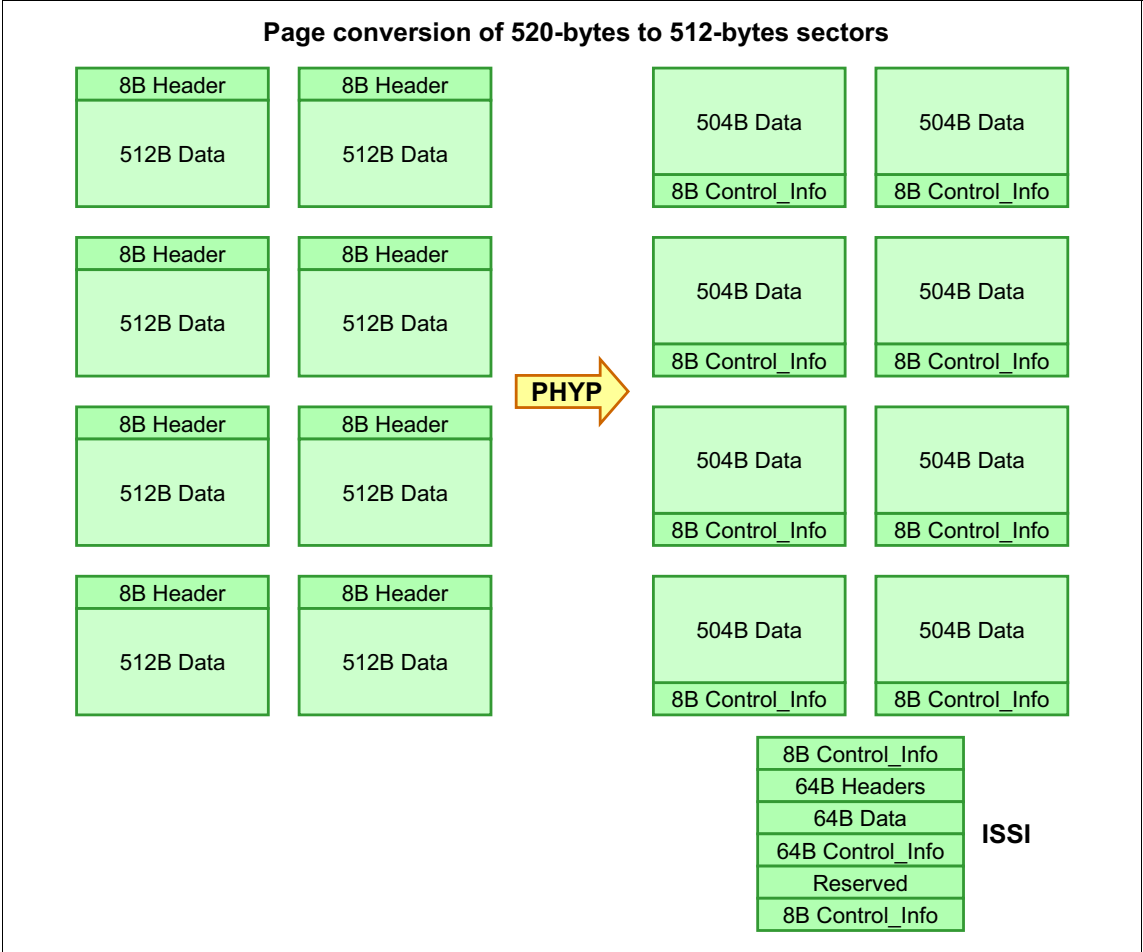


Figure 4-10 Page conversion of 520-bytes to 512-bytes sectors

Capacity: Due to the 8 to 9 sector conversion, the usable and reported capacity of virtual SCSI LUNs on IBM i, is only 8/9 of the configured storage capacity, that is, 11% less than the storage capacity configured for the Virtual I/O Server.

Each virtual SCSI LUN is reports in on the IBM i client as a generic virtual disk unit of type 6B22 model 050, under a virtual storage IOA and IOP type 290A representing the virtual SCSI client adapter as shown in Figure 4-11.

Logical Hardware Resources Associated with IOP				
Type options, press Enter.				
2=Change detail 4=Remove 5=Display detail 6=I/O debug				
7=Verify 8=Associated packaging resource(s)				
Opt	Description	Type-Model	Status	Resource Name
	Virtual IOP	* 290A-001	Operational	CMB01
	Virtual Storage IOA	290A-001	Operational	DC01
	Disk Unit	* 6B22-050	Operational	DD001
	Disk Unit	6B22-050	Operational	DD003
	Disk Unit	6B22-050	Operational	DD004
	Disk Unit	6B22-050	Operational	DD002
F3=Exit F5=Refresh F6=Print F8=Include non-reporting resources				
F9=Failed resources F10=Non-reporting resources				
F11=Display serial/part numbers F12=Cancel				

Figure 4-11 Virtual SCSI disk unit reported on IBM i

Support: Up to 16 virtual disk LUNs *and* up to 16 virtual optical LUNs are supported per IBM i virtual SCSI client adapter.

IBM i uses a queue depth of 32 per virtual SCSI disk unit and path, which is considerably larger when compared to the queue depths of 6, used for IBM i NPIV or native SAN storage attachment.

Tip: There is usually no need to be concerned about the larger IBM i queue depth. If the IBM i disk I/O response time shows a high amount of wait time as an indication for a bursty I/O behavior, consider using more LUNs or more paths to increase the IBM i I/O concurrency.

The IBM i virtual tape support by the Virtual I/O Server is slightly different when compared to the virtual SCSI support for disk storage devices. The IBM i client partition needs to know about the physical tape drive device characteristics, especially for Backup Recovery and Media Services (BRMS). This information is needed to determine, for example, which device class to use, and which tape drives (of the same device class) can be used for parallel saves/restores, for example, to avoid mixing together virtualized DAT and LTO drives.

Therefore, unlike virtual SCSI disk support, virtual tape support in the Virtual I/O Server provides a virtual SCSI special pass-through mode, to provide the IBM i client partition with the real device characteristics. The virtual LTO or DAT tape drive thus reports in on IBM i under a virtual storage IOP/IOA 29A0 with its native device type and model, such as 3580-004 for a LTO4 tape.

N_Port ID Virtualization

IBM i 6.1.1 or later, as a client of the PowerVM Virtual I/O Server, supports N_Port ID Virtualization (NPIV) through Virtual Fibre Channel for IBM System Storage DS8000 series and selected IBM System Storage tape libraries (see also “Requirements” on page 190).

Instead of emulated generic SCSI devices presented to the IBM i client partition by the Virtual I/O Server when using virtual SCSI, using NPIV uses the Virtual I/O Server acting as a Fibre Channel pass-through. This allows the IBM i client partition to see its assigned SCSI target devices, with all their device characteristics such as type and model information, as if they were natively attached, as shown in Figure 4-12.

Logical Hardware Resources Associated with IOP			
Type options, press Enter.			
2=Change detail 4=Remove 5=Display detail 6=I/O debug			
7=Verify 8=Associated packaging resource(s)			
Opt	Description	Type-Model	Status Resource Name
-	Virtual IOP	6B25-001	Operational CMB02
-	Virtual Storage IOA	6B25-001	Operational DC02
-	Disk Unit	2107-A85	Operational DD004
-	Disk Unit	2107-A85	Operational DD002
-	Tape Library	3584-032	Operational TAPMLB02
-	Tape Unit	3580-003	Operational TAP01
F3=Exit F5=Refresh F6=Print F8=Include non-reporting resources			
F9=Failed resources F10=Non-reporting resources			
F11=Display serial/part numbers F12=Cancel			

Figure 4-12 NPIV devices reported on IBM i

From this perspective, Virtual Fibre Channel support for IBM i is especially interesting for tape library attachment or DS8000 series attachment with IBM PowerHA® IBM SystemMirror® for i using DS8000 Copy Services storage-based replication for high availability or disaster recovery, for which virtual SCSI is not supported.

Virtual Fibre Channel allows sharing a physical Fibre Channel adapter between multiple IBM i partitions, to provide each of them native-like access to an IBM tape library. This avoids the need to move Fibre Channel adapters between partitions using the dynamic LPAR function.

IBM PowerHA SystemMirror for i with using DS8000 Copy Services fully supports Virtual Fibre Channel on IBM i. It even allows sharing a physical Fibre Channel adapter between different IBM i Independent Auxiliary Storage Pools (IASPs) or SYSBAS. Using Virtual Fibre Channel with PowerHA does not require dedicated Fibre Channel adapters for each SYSBAS and IASP anymore. This is because the IOP reset, which occurs when switching an IASP, affects the virtual Fibre Channel client adapter only. In a native-attached storage environment, switching an IASP will reset all the ports of the physical Fibre Channel adapter.

Multipathing and mirroring

The IBM i mirroring support for virtual SCSI LUNs, available since IBM i 6.1, was extended with IBM i 6.1.1 or later to support IBM i multipathing for virtual SCSI LUNs also.

Both IBM i multipathing and IBM i mirroring are also supported with Virtual Fibre Channel.

When using IBM i as a client of the Virtual I/O Server, consider using either IBM i multipathing or mirroring across two Virtual I/O Servers for redundancy, to protect the IBM i client from Virtual I/O Server outages, such as disruptive maintenance actions such as fix pack activations, as shown in Figure 4-13.

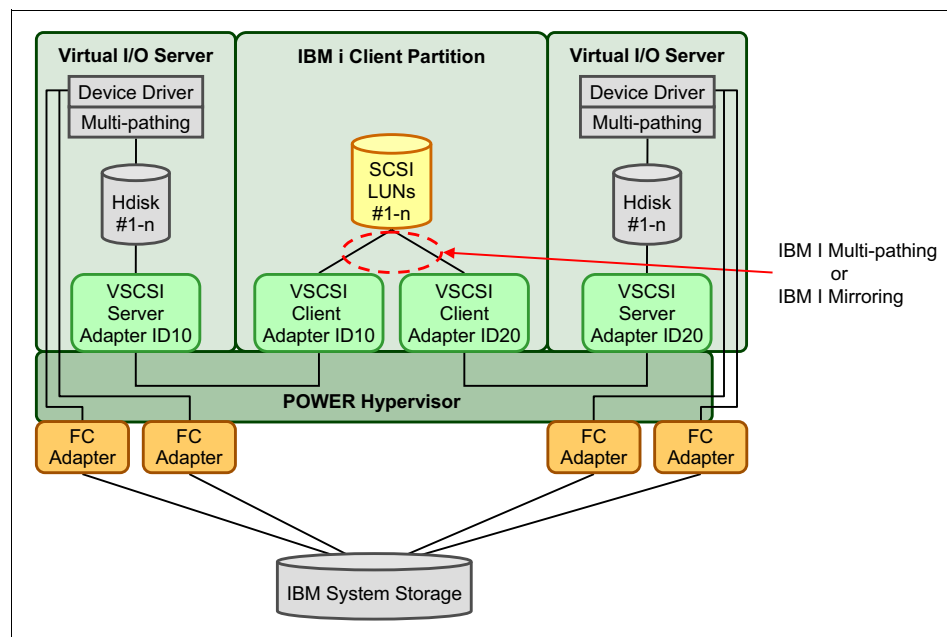


Figure 4-13 IBM i multipathing or mirroring for virtual SCSI

Both the IBM i mirroring and the IBM i multipathing function are fully integrated functions implemented in the IBM i System Licensed Internal Code storage management component. Unlike other Open System platforms, no additional device drivers are required to support these functions on IBM i.

With IBM i mirroring, disk write I/O operations are sent to each side, A and B, of a mirrored disk unit. For read I/O the IBM i mirroring algorithm selects the side to read from for a mirrored disk unit based on which side has the least amount of outstanding I/O. With equal storage performance on each mirror side, the read I/O is evenly spread across both sides of a mirrored disk unit.

IBM i multipathing supports up to 8 paths for each multipath disk unit. It uses a round-robin algorithm for load balancing to distribute the disk I/O across the available paths for each disk unit.

4.4.2 I/O virtualization consideration for Linux

Most of the PowerVM capabilities can be used by the supported versions of Linux. Linux can be installed in a dedicated or shared processor partition. Linux running in a partition can use physical devices and virtual devices. It can also participate in virtual Ethernet and can access external networks through Shared Ethernet Adapters (SEAs). A Linux partition can use virtual SCSI adapters and virtual Fibre Channel adapters.

The following terms and definitions are general:

- Virtual I/O client** Any partition that is using virtualized devices provided by other partitions.
- Virtual I/O Server** A special-function appliance partition that is providing virtualized devices to be used by client partitions.

Tools: For Linux on POWER systems, hardware service diagnostic aids and productivity tools, as well as installation aids for IBM servers running the Linux operating systems on POWER4 and later processors, are available from the IBM *Service and productivity* web page:

<http://www14.software.ibm.com/webapp/set2/sas/f/lopdiags/home.html>

Linux device drivers for IBM Power Systems virtual devices

IBM worked with Linux developers to create device drivers for the latest Linux kernel that enable Linux to use the IBM Power Systems virtualization features.

Table 2.1 shows all the kernel modules for IBM Power Systems virtual devices.

Table 4-1 Kernel modules for IBM Power Systems virtual devices

Linux kernel module	Supported virtual device	Source file locations, relative to /usr/src/linux/drivers/
hvc	virtual console server	char/hvc*
ibmveth	virtual Ethernet	net/ibmveth*
ibmvscsic	virtual SCSI - client/initiator	scsi/ibmvscsi*
ibmvfc	virtual Fibre Channel	scsi/ibmvscsi*

The latest Linux kernel source can be downloaded from this site:

<ftp://ftp.kernel.org/pub/linux/kernel/>

Precompiled Linux kernel modules are included with some Linux distributions.

Linux as Virtual I/O Server client

Linux running in a partition of an IBM Power Systems server can use virtual Ethernet adapters and virtual storage devices provided by Virtual I/O Servers.

Virtual console

IBM Power Systems provide a virtual console `/dev/hvc0` to each Linux partition.

Virtual Ethernet

To use virtual Ethernet adapters with Linux, the Linux kernel module `ibmveth` must be loaded. If IEEE 802.1Q VLANs are used, then, in addition, the Linux kernel module `8021q` must be available. Virtual Ethernet adapters use the same naming scheme such as physical Ethernet adapters, such as `eth0` for the first adapter. VLANs are configured by the **`vconfig`** command.

Linux can use inter-partition networking with other partitions and share access to external networks with other Linux and AIX partitions, for example, through a Shared Ethernet Adapter (SEA) of a PowerVM Virtual I/O Server.

Virtual SCSI client

The IBM virtual SCSI client for Linux is implemented by the `ibmvscsic` Linux kernel module. When this kernel module is loaded, it scans and auto-discovers any virtual SCSI disks provided by the Virtual I/O Servers.

Virtual SCSI disks will be named just as regular SCSI disks, for example, `/dev/sda` for the first SCSI disk or `/dev/sdb3` for the third partition on the second SCSI disk.

Virtual Fibre Channel client

The IBM virtual Fibre Channel client for Linux is implemented by the `ibmvfc` Linux kernel module. When this kernel module is loaded, it scans and auto-discovers any virtual Fibre Channel devices provided by the Virtual I/O Servers.

MPIO

Linux has support for generic and some vendor-specific implementations of Multipath I/O (MPIO), and some vendors provide additional MPIO-capable device drivers for Linux.

MPIO involving the Linux client can be configured in the following ways:

- ▶ MPIO access to the same disk through two Virtual I/O Servers.
- ▶ MPIO access to the same disk through one Virtual I/O Server that has two paths to the same disk.

MPIO single client and single Virtual I/O Server

Figure 4-14 shows how MPIO can be implemented in the Virtual I/O Server to provide redundant access to external disks for the virtual I/O client. However, implementing MPIO in the Virtual I/O Server instead of in the virtual I/O client does not provide the same degree of high availability to the virtual I/O client, because the virtual I/O client has to be shut down when the single Virtual I/O Server is brought down, for example, when the Virtual I/O Server is upgraded.

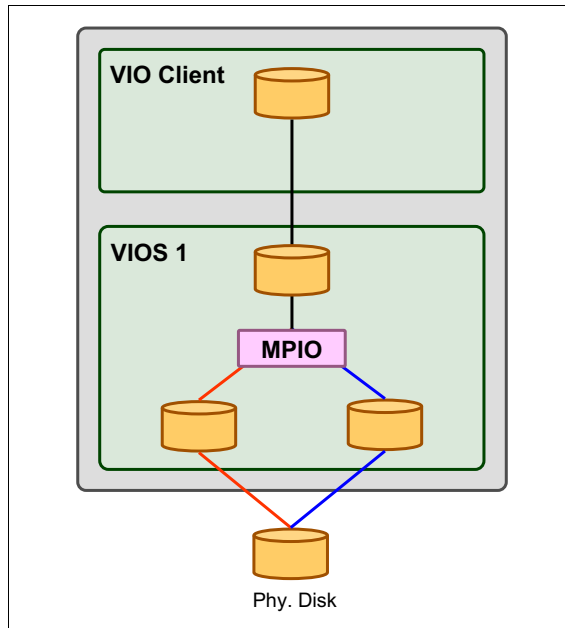


Figure 4-14 Single Virtual I/O Server with dual paths to the same disk

MPIO single client and dual Virtual I/O Server

MPIO can also be implemented using a single virtual Linux client and a dual Virtual I/O Servers where the two Virtual I/O Servers are accessing the same disks. This creates an environment of flexibility and reliability. In the event that one Virtual I/O Server is shut down, the virtual client can utilize the other Virtual I/O Server to access the single disk.

This capability is possible in SLES 9 and later as well as RHEL 5 and later. Red Hat 5 distributions require one boot parameter for this configuration to work correctly. The parameter “install mpath” must be added to the kernel boot line for the configuration shown in Figure 4-15 to work correctly. Starting with Red Hat 6, this parameter is no longer required.

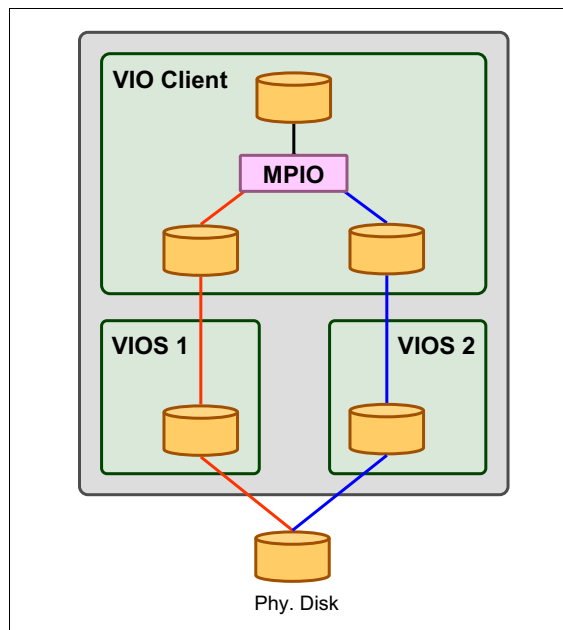


Figure 4-15 Dual Virtual I/O Server accessing the same disk

Mirroring

Linux can mirror disks by use of the RAID-Tools. Thus, for redundancy, you can mirror each virtual disk provided by one Virtual I/O Server to another virtual disk provided by a different Virtual I/O Server.

Implementing mirroring in a single Virtual I/O Server instead of mirroring the virtual I/O client storage within the client through two Virtual I/O Servers does not provide the same degree of high availability. This is because the virtual I/O client will lose its connection to the storage when the single Virtual I/O Server is upgraded or serviced.

The difference between mirroring in the virtual I/O client and in the Virtual I/O Server is shown in Figure 4-16.

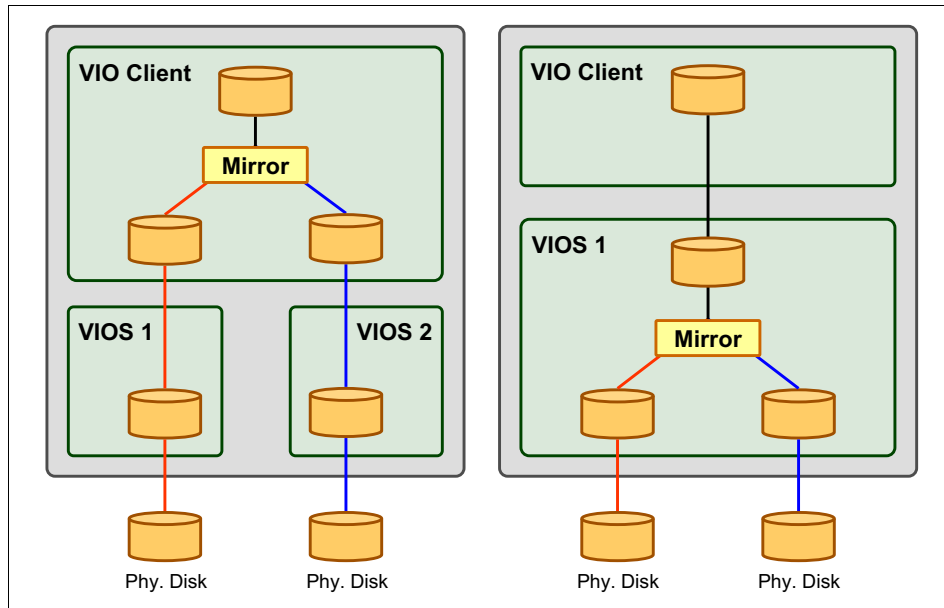


Figure 4-16 Implementing mirroring at client or server level

Considerations

The use of Linux as a virtual I/O client is subject to the following considerations:

- ▶ Only specific Linux distributions are supported as virtual I/O clients.
- ▶ Use of the Virtual I/O Server requires the purchase of the PowerVM.



Server virtualization overview

This chapter introduces the following PowerVM features:

- ▶ Live Partition Mobility
- ▶ Partition Suspend and Resume

5.1 Live Partition Mobility overview

PowerVM Live Partition Mobility allows you to move a running logical partition, including its operating system and running applications, from one system to another without any shutdown or without disrupting the operation of that logical partition. Inactive partition mobility allows you to move a powered off logical partition from one system to another. Suspended partition mobility allows moving a suspended partition from one system to another.

Figure 5-1 shows an example of Live Partition Mobility. Initiated from the HMC, we can move the client partition from the source server to the target server without disrupting the operating system and applications on the partition.

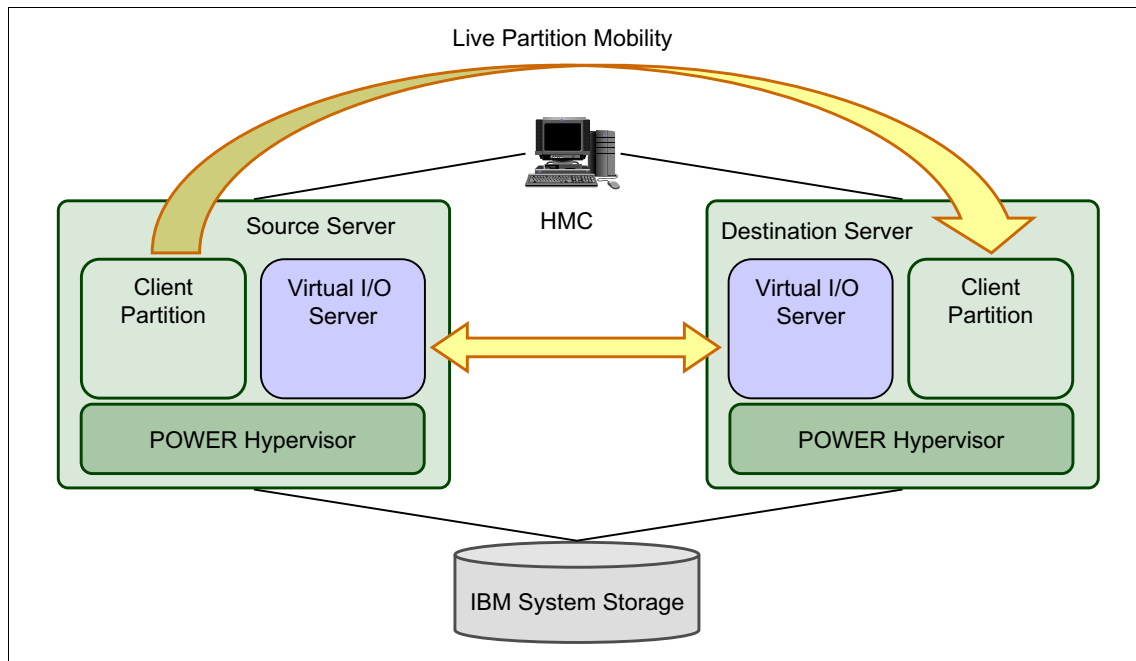


Figure 5-1 An Example for Live Partition Mobility

Partition mobility provides systems management flexibility and improves system availability, as follows:

- ▶ Avoid planned outages for hardware or firmware maintenance by moving logical partitions to another server and then performing the maintenance. Live Partition Mobility can help lead to zero downtime maintenance because you can use it to work around scheduled maintenance activities.
- ▶ Avoid downtime for a server upgrade by moving logical partitions to another server and then performing the upgrade. This allows your end users to continue their work without disruption.
- ▶ Perform preventive failure management: If a server indicates a potential failure, you can move its logical partitions to another server before the failure occurs. Partition mobility can help avoid unplanned downtime.
- ▶ Optimize server workloads:
 - Workload consolidation: You can consolidate workloads running on several small, under-utilized servers onto a single large server.
 - Flexible workload management: You can move workloads from server to server to optimize resource use and workload performance within your computing environment. With active partition mobility, you can manage workloads with minimal downtime.
- ▶ Use Live Partition Mobility for a migration from POWER6 and later POWER processor-based servers without any downtime of your applications.
- ▶ Using IBM Systems Director VMControl's system pool function, virtual server relocation using LPM can be automated, based on user defined policies or event triggers.

The migration manager function resides on the HMC and is in charge of configuring both systems. It has the responsibility of checking that all hardware and software prerequisites are met. It executes the required commands on the two systems to complete migration while providing migration status to the user.

Note: HMC Version 7 Release 3.4 introduces remote migration, the option of migrating partitions between systems managed by different HMCs. See “Remote Live Partition Mobility” on page 288 for details on remote migration.

When an inactive migration is performed, the HMC invokes the configuration changes on the two systems. For a suspended migration the partition's suspended state is transferred to the destination system. During an active migration, the running state (memory, registers, and so on) of the mobile partition is transferred during the process.

Memory management of an active migration is assigned to a mover service partition on each system. During an active partition migration, the source mover service partition extracts the mobile partition's state from the source system and sends it over the network to the destination mover service partition, which in turn updates the memory state on the destination system.

Any Virtual I/O Server partition can be configured as a mover service partition.

Live Partition Mobility has no specific requirements on the mobile partition's memory size or the type of network connecting the mover service partitions. The memory transfer is a process that does not interrupt a mobile partition's activity and might take time when a large memory configuration is involved on a slow network. Use a high bandwidth connection, such as 1 Gbps Ethernet or larger.

5.2 Partition Suspend and Resume overview

The Virtual I/O Server provides Partition Suspend and Resume capability to client logical partitions within the IBM POWER7 systems. Suspend/Resume operations allow the partition's state to be suspended and resumed at a later time.

A suspended logical partition indicates that it is in standby/hibernated state, and all of its resources can be used by other partitions. On the other hand, a resumed logical partition means that the partition's state has been successfully restored from a suspend operation. A partition's state is stored in a paging space on a persistent storage device.

The Suspend/Resume feature has been built on existing Logical Partition Mobility (LPM) and Active Memory Sharing (AMS) architecture, and it requires PowerVM Standard Edition.

Suspend capable partitions are available on POWER7 Systems and support the AIX operating system. For more details about supported hardware and operating systems, see Table 7-2 on page 95 and Table 7-4 on page 98.

The applicability and benefits of the Suspend/Resume feature include resource balancing and planned CEC outages for maintenance or upgrades. Lower priority and/or long running workloads can be suspended to free resources. This is useful for performance and energy management.

Suspend/Resume can be used in place of or in conjunction with Live Partition Mobility, and might require less time and effort than a manual database shutdown/restart.

Requirements: Suspend/Resume requires PowerVM *Standard Edition* (SE). However, when used in conjunction with Live Partition Mobility, it requires PowerVM *Enterprise Edition* (EE).

A typical scenario in which the Suspend/Resume capability is valuable is the case where a partition with a long running application can be suspended to allow for maintenance or upgrades and then resumed afterwards.

The availability requirements of the application might be such that configuring the partition for Partition Mobility is not warranted. However, the application does not provide its own periodic checkpoint capability, and shutting it down means restarting it from the beginning at a later time.

The ability to suspend processing for the partition, save its state safely, free up the system for whatever activities are required, and then resume it later, can be very valuable in this scenario.

Another example is the case where a partition is running applications that require 1-2 hours to safely shut them all down before taking the partition down for system maintenance and another 1-2 hours to bring them back up to steady state operation after the maintenance window.

Partition migration can be used to mitigate this scenario as well, but might require resources that are not available on another server. The ability to Suspend/Resume the partition in less time will save hours of administrator time in shutdown and startup activities during planned outage windows.

For more information about Partition Suspend and Resume, see sections 11.3, “Suspend and Resume planning” on page 297 and 17.2, “Suspend and Resume setup” on page 648.



Management console overview

This chapter describes the management consoles available for PowerVM:

- ▶ Hardware Management Console
- ▶ Integrated Virtualization Manager
- ▶ Systems Director VMControl

6.1 Management console comparison

Table 6-1 compares the features of PowerVM management consoles.

Table 6-1 PowerVM Management console comparison

Feature	HMC	IVM
Included in PowerVM		✓
Manage Power Blades		✓
Manage more than one server	✓	
Hardware monitoring	✓	✓
Service Agent call home	✓	✓
Graphical Interface	✓	✓
Requires a separate server to run on	✓	
Run on virtualized environment		
Advanced PowerVM features	✓	
High-end servers	✓	
Low-end and midrange servers	✓	✓ ^a
Servers families support	Power5/Power5+: ✓ Power6/Power6+: ✓ Power7: ✓	Power5/Power5+: ✓ Power6/Power6+: ✓ Power7: ✓
Redundant setup	✓	

a. Midrange for POWER7 technology-based servers is not supported.

6.2 Hardware Management Console

The Hardware Management Console (HMC) is a dedicated Linux-based appliance that you use to configure and manage IBM Power System servers. The HMC provides access to logical partitioning functions, service functions, and various system management functions through both a browser-based interface and a command line interface (CLI). Because it is a separate stand-alone system, the HMC does not use any managed system resources and you can maintain it without affecting system activity.

IBM PowerVM technology and HMC has been enhanced with the following features:

- ▶ Support up to 16 concurrent Live Partition Mobility (LPM) activities.
- ▶ DLPAR Add or Remove of virtual I/O adapters to or from a Virtual I/O Server (VIOS).

HMC will now automatically attempt to run the Add/Remove commands (cfgdev/rmdev) on the VIOS for the user. Prior to this enhancement, the user had to manually run these commands on the VIOS.

System port (virtual TTY/console support)

Each partition needs to have access to a system console. Tasks such as operating system install, network setup, and some problem analysis activities require a dedicated system console. Depending on the system configuration, the operating system console can be provided by the HMC or IVM virtual TTY.

For AIX and Linux, the POWER Hypervisor provides a virtual console using a virtual TTY or serial adapter and a set of POWER Hypervisor calls to operate on them.

For IBM i, an HMC managed server can use the 5250 system console emulation that is provided by a Hardware Management Console, or use an IBM i Access Operations Console. IVM managed servers must use an IBM i Access Operations Console.

Ports:

- ▶ The serial ports on an HMC-based system are inactive. Partitions requiring a TTY device must have an async adapter defined. The async adapter can be dynamically moved into or out of partitions with dynamic LPAR operations.
- ▶ On the IVM, the serial ports are configured and active. They are used for initial configuration of the system.

For smaller or segmented and distributed environments, not all functions of an HMC are required, and the deployment of an additional management server might not be suitable. IBM developed an additional hardware management solution called the Integrated Virtualization Manager (IVM) that provides a convenient browser-based interface and can perform a subset of the HMC functions. It was integrated into the Virtual I/O Server product that runs in a separate partition of the managed server itself, which avoids the need for a dedicated HMC server.

Because IVM itself is provided as a no cost option, it lowers the cost of entry into PowerVM virtualization. However, IVM can only manage a single Power System server. If IVM is the chosen management method, then a VIOS partition is defined and all resources belong to the VIOS. This means that no partition that is created can have dedicated adapters; instead, they are all shared.

6.3 Integrated Virtualization Manager

Integrated Virtualization Manager (IVM) is a management tool that combines partition management and Virtual I/O Server (VIOS) functionality into a single partition running on the system. The Integrated Virtualization Manager features an easy-to-use point-and-click interface and is supported on blades and entry-level to mid-range servers. Using the Integrated Virtualization Manager helps lower the cost of entry to PowerVM virtualization because it does not require a Hardware Management Console.

However, the Integrated Virtualization Manager can only manage a single Power System server. If Integrated Virtualization Manager is the chosen management method, then a Virtual I/O Server partition is defined and all resources belong to the Virtual I/O Server. This means that no partition that is created can have dedicated adapters; instead, they are all shared.

6.4 System Director VMControl

IBM Systems Director is a platform-management foundation that streamlines the way you manage physical and virtual systems across a heterogeneous environment. By using industry standards, IBM System Director supports multiple operating systems and virtualization technologies across IBM and non-IBM x86 platforms. IBM Systems Director has three editions: Express, Standard and Enterprise Editions.

IBM Systems Director VMControl is a plug-in for IBM Systems Director. It is a cross-platform suite of product that assists you in rapidly deploying virtual appliances to create virtual servers that are configured with the operating system and software applications that you want. It also enables you to group resources into system pools, which enables you to centrally manage and control the workloads in your environment.

The IBM System Director VMControl plug-in is available in four editions: VMControl Express Edition, VMControl Standard Edition, VMControl Enterprise Edition, and IBM System Director VMControl for IBM PowerLinux™, each with an increasing set of functionality for virtualization management as summarized next.

VMControl Express Edition includes features that were formerly part of IBM Systems Director virtualization manager. It enables you to complete the following tasks:

- ▶ Create virtual servers.
- ▶ Edit virtual servers.
- ▶ Manage virtual servers.
- ▶ Relocate virtual servers.
- ▶ Discover virtual server, storage, and network resources and visualize the physical-to-virtual relationships.

VMControl Standard Edition is a licensed feature of VMControl. With VMControl Standard Edition, you can complete the following tasks:

- ▶ Create new image repositories for storing virtual appliances and discover existing image repositories in your environment.
- ▶ Import external, standards-based virtual appliance packages into your image repositories as virtual appliances.
- ▶ Capture a running virtual server that is configured just the way you want, complete with guest operating system, running applications, and virtual server definition. When you capture the virtual server, a virtual appliance is created in one of your image repositories with the same definitions and can be deployed multiple times in your environment.
- ▶ Import virtual appliance packages that exist in the Open Virtualization Format (OVF) from the Internet or other external sources. After the virtual appliance packages are imported, you can deploy them within your data center.
- ▶ Deploy virtual appliances quickly to create new virtual servers that meet the demands of your ever-changing business needs.
- ▶ Create, capture, and manage workloads.

IBM Systems Director VMControl Enterprise Edition and *IBM Systems Director VMControl for IBM PowerLinux* are also licensed features of VMControl. With VMControl Enterprise Edition and IBM Systems Director VMControl for IBM PowerLinux, you can complete the following tasks:

- ▶ Create server system pools, which enable you to consolidate your resources and workloads into distinct and manageable groups.
- ▶ Deploy virtual appliances into server system pools.
- ▶ Manage server system pools, including adding hosts or additional storage space and monitoring the health of the resources and the status of the workloads in them.
- ▶ Group storage systems together using storage system pools to increase resource utilization and automation.
- ▶ Manage storage system pools by adding storage, editing the storage system pool policy, and monitoring the health of the storage resources.

For more information about VMControl, see the IBM Systems Director Information Center at this website:

<http://publib.boulder.ibm.com/infocenter/director/pubs/>



Part 2

Plan

In this part, we explain the planning requirements and considerations for PowerVM virtualization.

This part includes the following topics:

- ▶ PowerVM considerations
- ▶ Processor virtualization planning
- ▶ Memory virtualization planning
- ▶ I/O virtualization planning
- ▶ Server virtualization planning



PowerVM considerations

This chapter presents considerations on PowerVM system requirements, licensing, availability, security, and management consoles. It helps you to decide which version of PowerVM is most suitable for your needs.

It covers the following topics:

- ▶ System requirements
- ▶ Availability planning for PowerVM
- ▶ Security planning for PowerVM
- ▶ Management Console considerations

7.1 System requirements

The following sections present the hardware, licensing, and operating system requirements associated to each PowerVM feature available.

7.1.1 Hardware requirements

PowerVM features are supported on the majority of the Power Systems offerings, however, there are some exceptions. The *Availability of PowerVM features by Power Systems* models web page contains a summary of which features are available on which server models:

<http://www.ibm.com/systems/power/software/virtualization/editions/features.html>

For more detailed information, see Table 7-1.

Table 7-1 Virtualization features supported by POWER technology levels

Feature	POWER5	POWER6	POWER7
Virtual SCSI	Yes	Yes	Yes
Virtual Ethernet	Yes	Yes	Yes
Shared Ethernet Adapter	Yes	Yes	Yes
Virtual Fibre Channel	No	Yes	Yes
Virtual Tape	Yes	Yes	Yes
Logical partitioning	Yes	Yes	Yes
DLPAR I/O adapter add/remove	Yes	Yes	Yes
DLPAR processor add/remove	Yes	Yes	Yes
DLPAR memory add/remove	Yes	Yes	Yes
Micro-partitioning	Yes	Yes	Yes
Shared Dedicated Capacity	Yes ^a	Yes	Yes
Multiple Shared Processor Pools	No	Yes	Yes
Virtual I/O Server	Yes	Yes	Yes
Integrated Virtualization Manager	Yes	Yes	Yes
Suspend and resume	No	No	Yes

Feature	POWER5	POWER6	POWER7
Shared Storage Pools	No	Yes	Yes
Thin provisioning	No	Yes	Yes
Active Memory Sharing	No	Yes	Yes
Active Memory Deduplication	No	No	Yes ^b
Active Memory Mirroring	No	No	Yes ^c
Live Partition Mobility	No	Yes	Yes
Simultaneous multithreading	Yes ^d	Yes	Yes ^e
Active Memory Expansion	No	No	Yes
Capacity on Demand ³	Yes	Yes	Yes
AIX Workload Partitions	Yes	Yes	Yes

a. Only capacity from shutdown partitions can be shared.

b. Need firmware level 7.4, or later.

c. Need mid-tier and large-tier POWER7 Systems or later, including Power 770, 780, and 795.

d. POWER5 supports 2 threads.

e. POWER7 or later supports 4 threads.

Table 7-2 lists the various models of Power System servers and indicates which POWER technology is used.

Table 7-2 Server model to POWER technology level cross-reference

POWER5	POWER6	POWER7
7037-A50	7778-23X/JS23	8406-70Y/PS700
8844-31U/JS21	7778-43X/JS43	8406-71Y/PS701
8844-51U/JS21	7998-60X/JS12	8406-71Y/PS702
9110-510	7998-61X/JS22	7895-22X/Flex System p260 compute node
9110-51A	8203-E4A/520	7895-42X/Flex System p460 compute node
9111-285	8203-E8A/550	8202-E4B/720
9111-520	8234-EMA/560	8202-E4C/720
9113-550	9117-MMA	8205-E6B/740

POWER5	POWER6	POWER7
9115-505	9119-FHA/595	8205-E6C/740
9116-561	9125-F2A/575	8231-E2B/710
9117-570	9406-MMA/570	8231-E1C/710
9118-575	9407-M15/520	8231-E2C/730
9119-590	9407-M25/520	8233-E8B/750
9119-595	9407-M50/550	8236-EC8/755
9131-52A		8246-L1C/PowerLinux 7R1
9133-55A		8246-L1S/PowerLinux 7R1
9405-520		8246-L2C/PowerLinux 7R2
9406-520		8246-L2S/PowerLinux 7R2
9406-525		9117-MMB/770
9406-550		9117-MMC/770
9406-570		9117-MMD/770
9406-590		9119-FHB/795
9406-595		9125-F2C/775
9407-515		9179-MHB/780
		9179-MHC/780
		9179-MHD/780

7.1.2 Licensing requirements

Before using PowerVM, you also need to have a valid license, or we can say feature code. You need to check your order and confirm that the feature code is included. If you order an upgrade feature, you need to input and activate the code first. The procedure to activate the code is described in 1.2.3, “Activating the PowerVM feature” on page 12.

Table 7-3 is an overview of the PowerVM feature codes on IBM Power Systems.

Table 7-3 PowerVM feature code overview

Type and model	Express Edition	Standard Edition ^a	Enterprise Edition ^b
7895-22X	5225	5227	5228
7895-23X	5225	5227	5228
7895-42X	5225	5227	5228
8202-E4B	5225	5227	5228
8202-E4C	5225	5227	5228
8205-E6C	5225	5227	5228
8205-E6B	5225	5227	5228
8231-E2B	5225	5227	5228
8231-E1C and 8231-E2C	5225	5227	5228
8233-E8B	7793	7794	7795
9117-MMB	Not offered	7942	7995
9117-MMC	Not offered	7942	7995
9117-MMD	Not offered	7942	7995
9119-FHB	Not offered	7943	8002
9179-MHB	Not offered	7942	7995
9179-MHC	Not offered	7942	7995
9179-MHD	Not offered	7942	7995

a. The feature codes for the Standard Edition provide all functions supplied with the Express Edition.

b. The feature codes for the Enterprise Edition provide all functions supplied with the Standard Edition.

7.1.3 Operating system requirements

Table 7-4 here summarizes the PowerVM features supported by the operating systems compatible with Power Systems technology. Using this table, combined with Table 7-1 on page 94 and Table 7-2 on page 95, you can determine the suitable operating system and hardware combination required for a given feature.

Table 7-4 Virtualization features supported by AIX, IBM i and Linux

Feature	AIX 5.3	AIX 6.1	AIX 7.1	IBM i 6.1.1	IBM i 7.1	RHEL 5.8	RHEL 6.3	SLES 10 SP4	SLES 11 SP3
Virtual SCSI	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Virtual Ethernet	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Shared Ethernet Adapter	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Virtual Fibre Channel	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Virtual Tape	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Logical partitioning	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
DLPAR I/O adapter add/remove	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
DLPAR processor add/remove	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
DLPAR memory add	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
DLPAR memory remove	Yes	Yes	Yes	Yes	Yes	No	Yes	No	Yes
Micro-partitioning	Yes	Yes	Yes	Yes	Yes	Yes ^a	Yes ^b	Yes ^a	Yes
Shared dedicated capacity	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Multiple Shared Processor Pools	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Virtual I/O Server	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Integrated Virtualization Manager	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Suspend and resume	No	Yes	Yes	No	Yes ^c	Yes	Yes	No	No
Shared Storage Pools	Yes	Yes	Yes	Yes	Yes ^d	Yes	Yes	Yes	No
Thin provisioning	Yes	Yes	Yes	Yes ^e	Yes ^e	Yes	Yes	Yes	No

Feature	AIX 5.3	AIX 6.1	AIX 7.1	IBM i 6.1.1	IBM i 7.1	RHEL 5.8	RHEL 6.3	SLES 10 SP4	SLES 11 SP3
Active Memory Sharing	No	Yes	Yes	Yes	Yes	No	Yes	No	Yes
Active Memory Deduplication	No	Yes ^f	Yes ^g	No	Yes ^h	No	Yes	No	Yes
Live Partition Mobility ⁱ	Yes	Yes	Yes	No	Yes ^{j k}	Yes	Yes	Yes	Yes
Simultaneous multithreading (SMT)	Yes ^l	Yes ^m	Yes	Yes ⁿ	Yes	Yes ^l	Yes	Yes ^l	Yes
Active Memory Expansion	No	Yes ^o	Yes	No	No	No	No	No	No
Capacity on Demand ^p	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
AIX Workload Partitions	No	Yes	Yes	No	No	No	No	No	No

a. This version can only support 10 virtual machines per core.

b. Need RHEL 6.3 Errata upgrade to support 20 virtual machines per core.

c. Requires IBM i 7.1 TR2 with PTF SI39077 or later.

d. Requires IBM i 7.1 TR1.

e. Will become fully provisioned device when used by IBM i.

f. Requires AIX 6.1 TL7 or later.

g. Requires AIX 7.1 TL1 or later.

h. Requires IBM i 7.1.4 or later.

i. To determine the recommended levels please see

<http://www14.software.ibm.com/webapp/set2/flrt/home#toggle>

j. Requires IBM i 7.1 TR4 PTF group or later. You can access this link for more details:

http://www-912.ibm.com/s_dir/SLKBase.nsf/1ac66549a21402188625680b0002037e/e1877ed7f3b0cfa8862579ec0048e067?OpenDocument#_Section1

k. Not supported on IVM.

l. Only supports two threads.

m. AIX 6.1 up to TL4 SP2 only supports two threads, and supports four threads as of TL4 SP3.

n. IBM i 6.1.1 and up support SMT4.

o. On AIX 6.1 with TL4 SP2 and later.

p. Available on selected models.

7.2 Availability planning for PowerVM

Because individual Power Systems offerings are capable of hosting many system images, the importance of isolating and handling service interruptions becomes greater. These service interruptions can be planned or unplanned. Carefully consider interruptions for systems maintenance when planning system maintenance windows, as well as other factors such as these:

- ▶ Environmental, including cooling and power
- ▶ System firmware
- ▶ Operating systems, for example, AIX, IBM i and Linux
- ▶ Adapter microcode

Technologies such as Live Partition Mobility or clustering (for example, IBM PowerHA System Mirror) can be used to move workloads between machines, allowing for scheduled maintenance, minimizing any service interruptions.

For applications requiring near-continuous availability, use clustering technology such as IBM PowerHA System Mirror to provide protection across physical machines. Locate these machines so that they are not reliant on any single support infrastructure element (for example, the same power and cooling facilities). In addition, consider environmental factors such as earthquake zones or flood plains.

The Power Systems servers, based on POWER technology, build upon a strong heritage of systems designed for industry-leading availability and reliability.

IBM takes a holistic approach to systems reliability and availability—from the microprocessor, which has dedicated circuitry and components designed into the chip, to Live Partition Mobility and the ability to move running partitions from one physical server to another. The extensive component, system, and software capabilities, which focus on reliability and availability, coupled with good systems management practice, can deliver near-continuous availability.

The base reliability of a computing system is, at its most fundamental level, dependent upon the design and intrinsic reliability of the components that comprise it. Highly reliable servers, such as Power Systems offerings, are built with highly reliable components. Power Systems technology allows for redundancies of several system components and mechanisms that diagnose and handle special situations, such as errors or failures at the component level.

Besides of the availability design on hardware, you can consider the following technologies to enhance your virtualization environment availability:

- ▶ Redundant Virtual I/O Servers
- ▶ Live Partition Mobility
- ▶ PowerVM with PowerHA

The following sections present more information on these technologies.

7.2.1 Redundant Virtual I/O Servers

Because an AIX, IBM i, or Linux partition can be a client of one or more Virtual I/O Servers at the same time, a good strategy to improve availability for sets of client partitions is to connect them to two Virtual I/O Servers.

Attention: IVM does not support redundant Virtual I/O Servers.

In a dual Virtual I/O Server configuration, virtual SCSI and Shared Ethernet Adapter can be configured in a redundant fashion allowing system maintenance such as reboot, software updates or even reinstallation to be performed on a Virtual I/O Server without affecting the virtual I/O clients. This is the main reason to implement two Virtual I/O Servers.

A basic architecture for dual Virtual I/O Servers is shown in Figure 7-1.

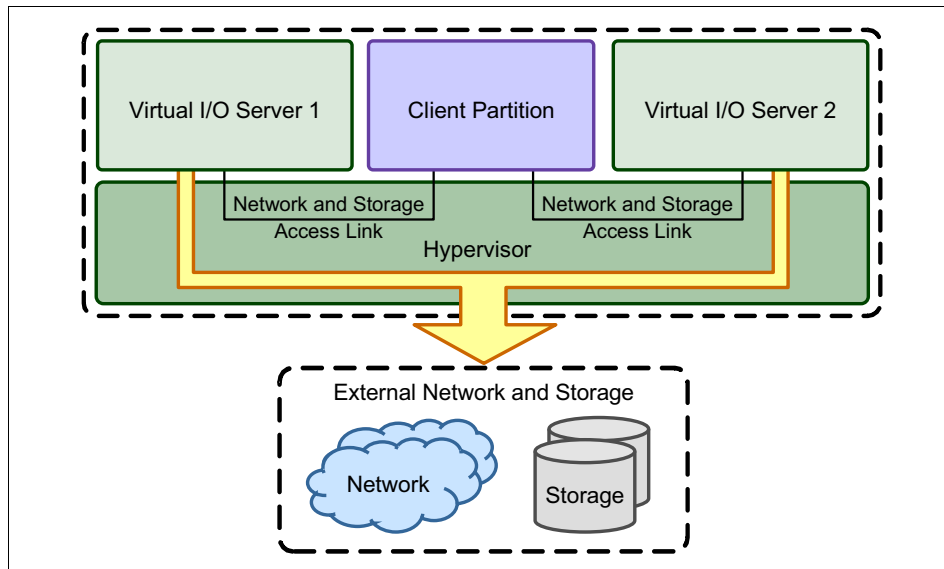


Figure 7-1 Sample for dual Virtual I/O Servers architecture

In Figure 7-1, the client partition accesses external network and storage resources through both Virtual I/O Servers. With the proper planning and architecture implementation, one Virtual I/O Server failure has no impact to the client partition. It also helps on the system maintenance. With the client partition using multipathing and SEA failover, no actions will need to be performed on the client partition while the system maintenance is being performed or after it has completed. This results in improved uptime and reduced system administration efforts for the client partitions.

For more information on SEA failover, see 16.3.2, “SEA failover” on page 591.

Scenario: Redundant Virtual I/O Servers can help to reduce and avoid both planned and unplanned outages.

7.2.2 Live Partition Mobility

An environment that has only small windows for scheduled downtime may use Live Partition Mobility to manage many scheduled activities either to reduce downtime through inactive migration or to avoid service interruption through active migration.

For example, if a system has to be shut down due to a scheduled power outage, its hosted partitions may be migrated to powered systems before the outage.

An example is shown in Figure 7-2, where system A has to be shut down. The production database partition is actively migrated to system B, while the production Web application partition is actively migrated to system C. The test environment is not considered vital and is shut down during the outage.

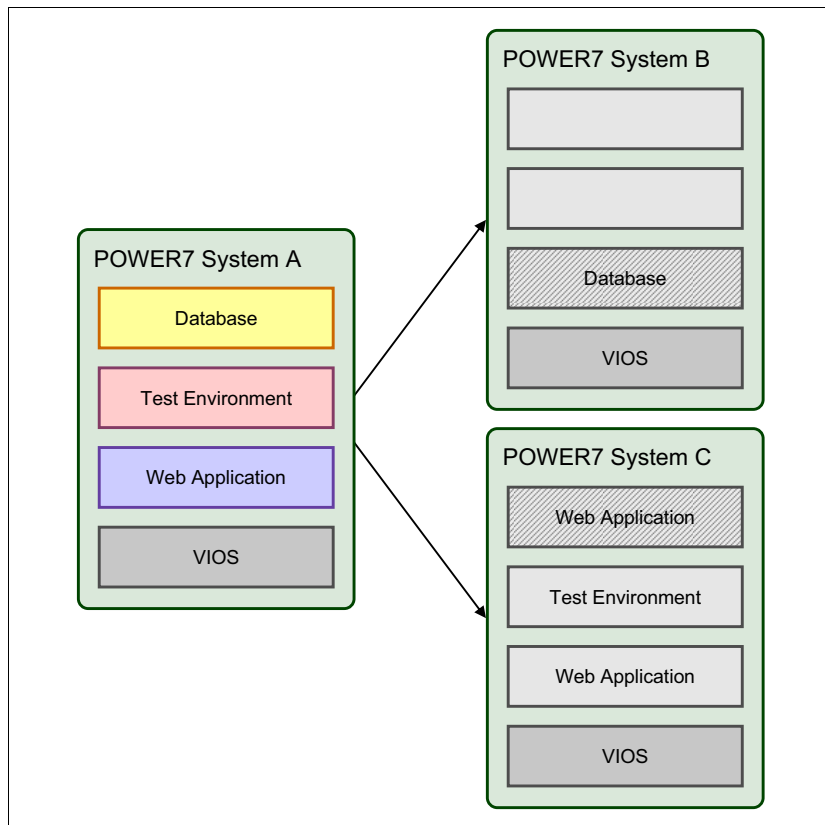


Figure 7-2 Migrating all partitions of a system

Live Partition Mobility is a reliable procedure for system reconfiguration and it may be used to improve the overall system availability. Live Partition Mobility increases global availability, but it is not a high availability solution to avoid the unplanned outage. It requires both source and destination systems to be operational and that the partition is not in a failed state. In addition, it does not monitor operating system and application state and it is, by default, an user-initiated action.

Scenario: Live Partition Mobility can only help to reduce the planned outage.

7.2.3 PowerVM with PowerHA

PowerHA is the IBM important high availability solution for UNIX (AIX) and IBM i systems. PowerHA can work with PowerVM to achieve higher availability. Figure 7-3 shows a typical architecture for using PowerHA in virtualization environment.

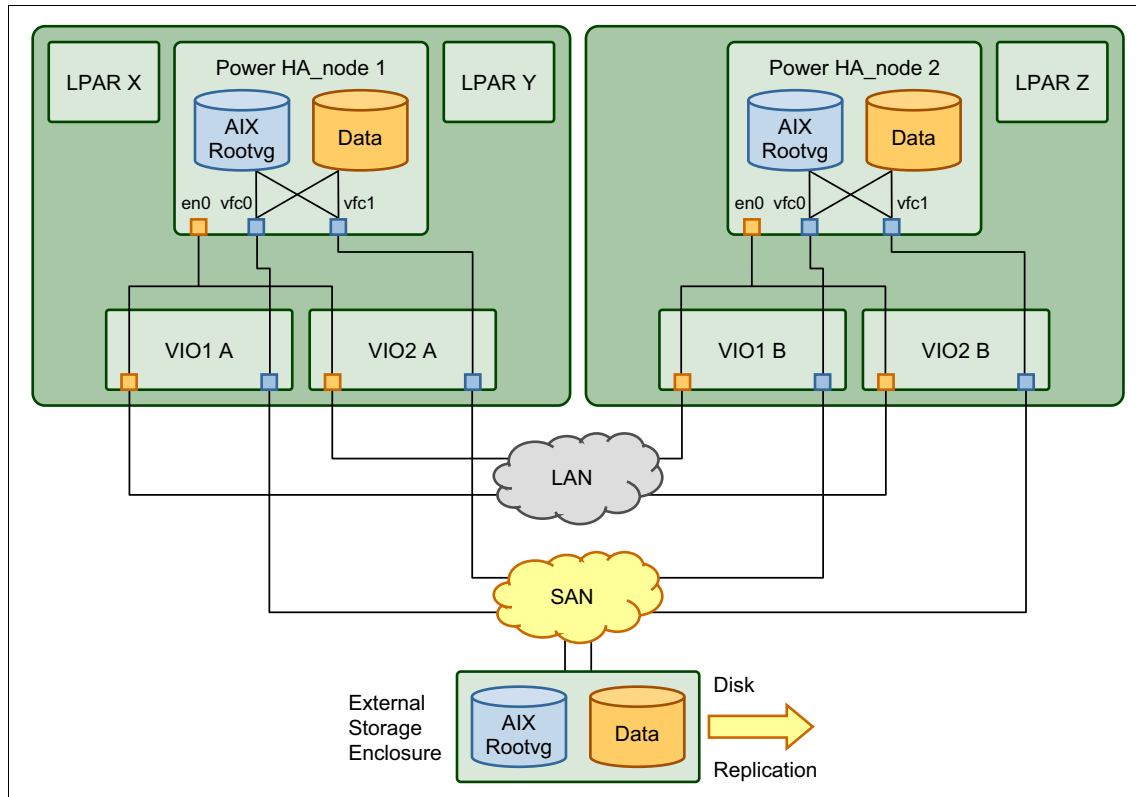


Figure 7-3 Live Partition Mobility capable virtualized environment diagram

In this case, you can use Live Partition Mobility or PowerHA to migrate or move the application from one system to another to avoid any planned outage. On the other hand, PowerHA monitors the running status for all the nodes of the cluster; in our case, the nodes PowerHA_node 1 and PowerHA_node 2. If one node failed due to some reason, PowerHA can realize it and trigger a series of actions, and bring the workload up on another available node based on the configured policy. That helps you to reduce the unplanned outage of the system.

For more details about PowerHA, you can read the book *Exploiting IBM PowerHA SystemMirror Enterprise Edition*, SG24-7841, available at the following website:

<http://www.redbooks.ibm.com/abstracts/sg247841.html>

You can also access the following site for more information about how to use PowerHA in PowerVM environment.

<http://pic.dhe.ibm.com/infocenter/aix/v7r1/index.jsp?topic=%2Fcom.ibm.aix.powerha.navigation%2Fpowerha.htm>

Scenario: PowerHA can help to reduce and avoid both planned and unplanned outages.

7.3 Security planning for PowerVM

Security and compliance are intrinsic to today's business processes, development and daily operations and should be factored in to the initial design of any IT or critical infrastructure solution, not bolted on after the fact.

In POWER Systems, all resources are controlled by the POWER Hypervisor. The Hypervisor ensures that any partition attempting to access resources within the system has permission to do so.

PowerVM has introduced a number of technologies allowing partitions to securely communicate within a physical system. To maintain the complete isolation of partition resources, the POWER Hypervisor enforces communications standards as normally applied to external infrastructure communications. For example, the virtual Ethernet implementation is based on the IEEE 802.1Q standard.

Another security enhancement on PowerVM is PowerSC. PowerSC is a suite of features that includes Security and Compliance Automation, Trusted Boot, Trusted Firewall, Trusted Logging, and Trusted Network Connect and Patch management. The security technology that is placed within the virtualization layer provides additional security to the standalone systems.

The PowerSC feature includes two editions: PowerSC Express Edition and PowerSC Standard Edition. Table 7-5 provides details about the editions, the features included in the editions, the components, and the processor-based hardware on which each component is available.

Table 7-5 PowerSC components, editions, and hardware support

Components	Description	Editions	Hardware Support
Security and Compliance Automation	Automates the setting, monitoring, and auditing of security and compliance configuration for Payment Card Industry Data Security Standard (PCI DSS), Sarbanes-Oxley Act and COBIT compliance (SOX/COBIT), and U.S. Department of Defense (DoD) Security Technical Implementation Guide (STIG).	PowerSC Express Edition PowerSC Standard Edition	POWER5 POWER6 POWER7
Trusted Boot	Measures the boot image, operating system, and applications, and attests their trust by using the virtual Trusted Platform Module (TPM) technology.	PowerSC Standard Edition	POWER7 firmware eFW7.4, or later
Trusted Firewall	Saves time and resources by enabling direct routing across specified Virtual LANs (VLANs) that are controlled by the same Virtual I/O Server.	PowerSC Standard Edition	POWER6 POWER7 Virtual I/O Server Version 6.1S, or later
Trusted Logging	The logs of AIX are centrally located on the Virtual I/O Server in real time. This feature provides tamperproof logging and convenient log backup and management.	PowerSC Standard Edition	POWER5 POWER6 POWER7
Trusted Network Connect and patch management	Verifies that all AIX systems in the virtual environment are at the specified software and patch level and provides management tools to ensure that all AIX systems are at the specified software level. Provides alerts if a down-level virtual system is added to the network or if a security patch is issued that affects the systems.	PowerSC Standard Edition	POWER5 POWER6 POWER7

You can find more information about PowerSC at the following sites:

- ▶ IBM PowerSC:
<http://www-03.ibm.com/systems/power/software/security/>
- ▶ IBM Information Center:
http://pic.dhe.ibm.com/infocenter/aix/v7r1/topic/com.ibm.aix.doc/doc/base/powersc_main.htm

Notice: PowerSC does not support the LIBM i and Linux platforms.

The Power Systems virtualization architecture, AIX, IBM i, and some Linux operating systems have been security certified to the EAL4+ level. For more information, see this website:

<http://www.ibm.com/systems/power/software/security/offerings.html>

7.4 Management Console considerations

You can choose to use HMC, IVM, or VMControl as the console to manage your PowerVM environment.

7.4.1 Hardware Management Console (HMC)

HMC is the most common console you can use to manage your PowerVM environment. If your requirements match the following criteria, you may consider to use HMC as your management console:

- ▶ You want to manage IBM Power 770, Power 780 and Power 795 machines.
- ▶ You want to manage more than one server through your management console.
- ▶ You need to manage dual Virtual I/O Servers environment.
- ▶ You need high availability for management console.

You can get more information about Hardware Management Console through the following link:

http://pic.dhe.ibm.com/infocenter/powersys/v3r1m5/topic/ipha8/hwparent_hmc.htm

And you can access the IBM Fix Central through the following link to check the latest firmware version for HMC:

<http://www.ibm.com/support/fixcentral/>

7.4.2 Integrated Virtualization Manager (IVM)

The IVM, coupled with the advanced capabilities of Power Systems, offers a lower cost of entry into virtualization on IBM POWER Systems. It provides a management model for a single system. But it does not offer all of the HMC capabilities. If your requirements match the following criteria, you may consider to use IVM as your management console:

- ▶ You do not want an additional hardware to manage the system.
- ▶ You need to manage IBM Power Blade.
- ▶ You do not need Virtual I/O Server redundancy or management console redundancy.

For more information, see *Integrated Virtualization Manager for Power Systems Servers*, REDP-4061. You can download it through the following link:

<http://www.redbooks.ibm.com/redpieces/abstracts/redp4061.html>

7.4.3 IBM System Director VMControl

VMControl is packaged as a plug-in to IBM System Director. It is designed to simplify the management of virtual environments across multiple virtualization technologies and hardware platforms.

Besides of the functions IVM and HMC provide, VMControl has some additional features, such as inventory management, virtual images management, and virtual resource pools management.

If your requirements match the following criteria, you may consider to use VMControl as your management console:

- ▶ You need to manage multiple virtualization technologies and hardware platforms.
- ▶ You need some automatic functions to help you to perform some complex management tasks.

Notice: You should already have either IVM or HMC available in order to use VMControl as the management console.

For more information about VMControl, you can see the IBM Systems Director Information Center at this website:

<http://publib.boulder.ibm.com/infocenter/director/pubs/>

And you can also find the publication *IBM Systems Director VMControl Implementation Guide on IBM Power Systems*, SG24-7829 through the following link:

<http://www.redbooks.ibm.com/abstracts/sg247829.html>



Processor virtualization planning

This chapter helps with planning for processor virtualization by providing information about the following topics:

- ▶ Micro-partition capacity planning
- ▶ Shared-Storage Pools capacity planning
- ▶ Software licensing in a virtualized environment

8.1 Micro-partitioning capacity planning

Starting with servers based on POWER7+, micro-partitions can be created with a minimum of 0.05 processing units. Consequently, you can create a maximum of 20 micro-partitions per activated processor.

In contrast, dedicated-processor LPARs can only be allocated whole processors, so the maximum number of dedicated-processor LPARs in a system is equal to the number of physical activated processors.

It is important to point out that the maximum number of micro-partitions supported for your system might not be the most practical configuration. Based on production workload demands, the number of micro-partitions that your system needs to use might be less.

Partitions are created and orchestrated by the HMC or IVM. When you start creating a partition, you have to choose between a micro-partition and a dedicated processor LPAR.

When setting up a partition, you need to define the resources that belong to the partition, such as memory and I/O resources. For micro-partitions, you have to configure these additional attributes:

- ▶ Minimum, desired, and maximum *processing units of capacity*
- ▶ The processing sharing mode, either *capped* or *uncapped*
- ▶ Minimum, desired, and maximum *virtual processors*

These settings are the topics of the following sections.

8.1.1 Processing units of capacity

Processing capacity can be configured in fractions of 0.01 processors. Starting with servers based on POWER7+, the minimum amount of processing capacity that has to be assigned to a micro-partition is 0.05 processors.

On the HMC, processing capacity is specified in terms of *processing units*. The minimum capacity of 0.05 processors is specified as 0.05 processing units. To assign a processing capacity representing 75% of a processor, 0.75 processing units are specified on the HMC.

On a system with two processors, a maximum of 2.0 processing units can be assigned to a micro-partition. Processing units specified on the HMC are used to quantify the minimum, desired, and maximum amount of processing capacity for a micro-partition.

After a micro-partition is activated, processing capacity is usually referred to as capacity entitlement or entitled capacity.

A micro-partition is guaranteed to receive its capacity entitlement under all systems and processing circumstances.

8.1.2 Capped and uncapped mode

Micro-partitions have a specific processing mode that determines the maximum processing capacity given to them from their Shared-Processor Pool.

The processing modes are as follows:

Uncapped mode	The processing capacity can exceed the entitled capacity when resources are available in their Shared-Processor Pool and the micro-partition is eligible to run. Extra capacity is distributed on a weighted basis. You must specify the uncapped weight of each micro-partition when it is created.
Capped mode	The processing capacity given can never exceed the entitled capacity of the micro-partition.

If there is competition for additional processing capacity among several uncapped micro-partition, the POWER Hypervisor distributes unused processor capacity to the eligible micro-partition in proportion to each micro-partition's uncapped weight. The higher the uncapped weight of a micro-partition, the more processing capacity the micro-partition will receive.

The uncapped weight must be an integer from 0 to 255. The default uncapped weight for uncapped micro-partitions is 128. The unused processor capacity available in the Shared-Processor-Pool is distributed among all runnable uncapped micro-partitions, based on the uncapped weight proportion of each.

So additional capacity for an eligible uncapped micro-partition is computed by dividing its uncapped weight by the sum of the uncapped weights for all uncapped partitions that are currently runnable in the dispatch window.

As an example, consider a case where there are 200 units of unused processing capacity available for reallocation to eligible micro-partitions ($200 = 2.0$ processors). The uncapped weighting of the micro-partition that is the subject to this calculation is 100. There are 5 runnable uncapped micro-partitions competing for the unused processor capacity, for which the weighting sum is 800. From this data, we can compute the additional capacity share:

$$\text{AdditionalCapacityShare} = 200 \times \frac{100}{800}$$

This gives us the following result:

$$\text{AdditionalCapacityShare} = 25$$

In this example, the AdditionalCapacityShare of 25 equates to 0.25 processor units.

Important: If you set the uncapped weight at 0, the POWER Hypervisor treats the micro-partition as a capped micro-partition. A micro-partition with an uncapped weight of 0 cannot be allocated additional processing capacity above its entitled capacity.

A weight of 0 allows automated workload management software to provide the equivalent function as a dynamic LPAR operation to change uncapped to capped (and the reverse).

8.1.3 Virtual processors

A virtual processor is a depiction or a representation of a physical processor that is presented to the operating system running in a micro-partition. The processing entitlement capacity assigned to a micro-partition, be it a whole or a fraction of a processing unit, will be distributed by the server firmware equally between the virtual processors within the micro-partition to support the workload. For example, if a micro-partition has 1.60 processing units and two virtual processors, each virtual processor will have the capacity of 0.80 processing units.

A virtual processor cannot have a greater processing capacity than a physical processor. The capacity of a virtual processor will be equal to or less than the processing capacity of a physical processor.

Selecting the optimal number of virtual processors depends on the workload in the partition. The number of virtual processors can also have an impact on software licensing, for example, if the sub-capacity licensing model is used. 8.3, “Software license in a virtualized environment” on page 133 describes licensing in more detail.

By default, the number of processing units that you specify is rounded up to the minimum whole number of virtual processors needed to satisfy the assigned number of processing units. The default settings maintain a balance of virtual processors to processor units. For example:

- ▶ If you specify 0.50 processing units, one virtual processor will be assigned.
- ▶ If you specify 2.25 processing units, three virtual processors will be assigned.

You can change the default configuration and assign more virtual processors in the partition profile.

A micro-partition must have enough virtual processors to satisfy its assigned processing capacity. This capacity can include its entitled capacity and any additional capacity above its entitlement if the micro-partition is uncapped.

So, the upper boundary of processing capacity in a micro-partition is determined by the number of virtual processors that it possesses. For example, if you have a partition with 0.50 processing units and one virtual processor, the partition cannot exceed 1.00 processing units. However, if the same partition with 0.50 processing units is assigned two virtual processors and processing resources are available, the partition can then use an additional 1.50 processing units.

The minimum number of processing units that can be allocated to each virtual processor is dependent on the server model. The maximum number of processing units that can be allocated to a virtual processor is always 1.00. Additionally, the number of processing units cannot exceed the total processing unit within a Shared-Processor Pool.

Number of virtual processors

In general, the value of the minimum, desired, and maximum virtual processor attributes needs to parallel those of the minimum, desired, and maximum capacity attributes in some fashion. A special allowance has to be made for uncapped micro-partitions, because they are allowed to consume more than their capacity entitlement.

If the micro-partition is uncapped, the administrator might want to define the desired and maximum virtual processor attributes greater than the corresponding capacity entitlement attributes. The exact value is installation-specific, but 50 to 100 percent more is reasonable.

Table 8-1 shows several reasonable settings for the number of virtual processors, processing units, and the capped and uncapped mode.

Table 8-1 Reasonable settings for micro-partitions

Min VPs ^a	Desired VPs	Max VPs	Min PU ^b	Desired PU	Max. PU	Capped
1	2	4	0.1	2.0	4.0	Y
1	3 or 4	8	0.1	2.0	8.0	N
1	2	6	0.1	2.0	6.0	Y
1	3 or 4	10	0.1	2.0	10.0	N

a - Virtual processors

b - Processing units

Virtual processor folding

In order for an uncapped micro-partition to take full advantage of unused processor capacity in the Physical Shared-Processor Pool, it must have enough virtual processors defined. In the past, these additional virtual processors could remain idle for substantial periods of time and consume a small but valuable amount of resources.

Virtual processor folding effectively puts idle virtual processors into a hibernation state so that they do not consume any resources. There are several important benefits of this feature including improved processor affinity, reduced POWER Hypervisor workload, and increased average time a virtual processor executes on a physical processor.

Following are the characteristics of the virtual processor folding feature:

- ▶ Idle virtual processors are not dynamically removed from the partition. They are *hibernated*, and only awoken when more work arrives.
- ▶ There is no benefit from this feature when partitions are busy.
- ▶ If the feature is turned off, all virtual processors defined for the partition are dispatched to physical processors.
- ▶ Virtual processors having attachments, such as **bindprocessor** or **rset** command attachments in AIX, are not excluded from being disabled.
- ▶ The feature can be turned off or on; the default is on.

When a virtual processor is disabled, threads are not scheduled to run on it unless a thread is bound to that processor.

Virtual processor folding is controlled through the *vpm_xvcpus* tuning setting, which can be configured using the **schedo** command.

For more information about virtual processor folding, including usage examples, hardware and software requirements see the IBM EnergyScale™ for POWER7 Processor-Based Systems white paper, which can be found at this website:

<http://www.ibm.com/systems/power/hardware/whitepapers/energyscale7.html>

8.1.4 Shared processor considerations

Take the following considerations into account when implementing the micro-partitions:

- ▶ Starting with servers based on POWER7+, the minimum size for a micro-partition is 0.05 processing units of a physical processor. So the number of micro-partitions you can activate for a system depends mostly on its number of activated processors.
- ▶ The maximum number of virtual processors in a micro-partition is 64.
- ▶ The minimum number of processing units you can have for each virtual processor depends on the server model. The maximum number of processing units that you can have for each virtual processor is always 1.00. This means that a micro-partition cannot use more processing units than the number of virtual processors that it is assigned, even if the micro-partition is uncapped.
- ▶ A partition is either a dedicated-processor partition or a micro-partition, it cannot be both. However, processor capacity for a micro-partition can come from Shared Dedicated Capacity. This is unused processor capacity from processors dedicated to a partition but that are capable of capacity donation. This situation does not change the characteristics of either the DEDICATED-processor partition or the micro-partition.
- ▶ If you want to dynamically remove a virtual processor, you cannot select a specific virtual processor to be removed. The operating system will choose the virtual processor to be removed.
- ▶ AIX, IBM i and Linux will utilize affinity domain information provided by firmware (POWER Hypervisor) to build associations of virtual processors to memory, and it will continue to show preference to redispaching a thread to the virtual processor that it last ran on. However, this cannot be guaranteed in all circumstances.
- ▶ An uncapped micro-partition with a weight of 0 is effectively the same as a micro-partition that is capped. This is because it will never receive any additional capacity above it's capacity entitlement. Using the HMC or IVM, the weighting of a micro-partition can be dynamically changed. Similarly, the HMC or IVM can change the mode of a micro-partition from capped to uncapped (and the reverse).

Dedicated processors

Dedicated processors are whole processors that are assigned to dedicated-processor partitions (LPARs). The minimum processor allocation for an LPAR is one (1) whole processor, and can be as many as the total number of installed processors in the server.

Each processor is wholly dedicated to the LPAR. It is not possible to mix shared processors and dedicated processors in the same partition.

By default, the POWER Hypervisor will make the processors of a powered-off LPAR available to the Physical Shared-Processor Pool. When the processors are in the Physical Shared-Processor Pool, an uncapped partition that requires more processing resources can utilize the additional processing capacity. However, when the LPAR is powered on, it will regain the processors and they will become dedicated to the newly powered-on LPAR.

To prevent dedicated processors from being used in the Physical Shared-Processor Pool while they are not part of a powered-on LPAR, you can disable this function on the HMC by deselecting the “Processor Sharing: Allow when partition is inactive” check box in the partition’s properties.

Attention: The option “Processor Sharing: Allow when partition is inactive” is activated by default. It is not part of profile properties and it cannot be changed dynamically.

Tip: If the “Processor Sharing: Allow when partition is inactive” box is checked on the HMC and you want to lock the dedicated processors from being released to the Physical Shared-Processor Pool without fully activating an operating system in a partition, you can do one of the following actions:

- ▶ For AIX and Linux, boot the partition to SMS.
- ▶ For IBM i, boot the partition to Dedicated Services Tools (DST).

Doing this will hold the processors and stop them from being included in the Physical Shared-Processor Pool.

8.2 Shared-Processor Pools capacity planning

This section describes the capacity attributes of Shared-Processor Pools and provides examples of the capacity resolution according to the server load.

8.2.1 Capacity attributes

The following attributes are used for calculating the pool capacity of Shared-Processor Pools:

- ▶ **Maximum Pool Capacity:**

Each Shared-Processor Pool has a maximum capacity associated with it. The Maximum Pool Capacity (MPC) defines the upper boundary of the processor capacity that can be utilized by the set of micro-partitions in the Shared-Processor Pool. The Maximum Pool Capacity must be represented by a whole number of processor units.

- ▶ **Reserved Pool Capacity:**

The system administrator can assign an entitled capacity to a Shared-Processor Pool for the purpose of reserving processor capacity from the Physical Shared-Processor Pool for the express use of the micro-partitions in the Shared-Processor Pool. The Reserved Pool Capacity (RPC) is in addition to the processor capacity entitlements of the individual micro-partitions in the Shared-Processor Pool. The Reserved Pool Capacity is distributed among uncapped micro-partitions in the Shared-Processor Pool according to their uncapped weighting. Default value for the Reserved Pool Capacity is zero (0).

- ▶ **Entitled Pool Capacity:**

The Entitled Pool Capacity (EPC) of a Shared-Processor Pool defines the guaranteed processor capacity that is available to the group of micro-partitions in the Shared-Processor Pool. The Entitled Pool Capacity is the sum of the entitlement capacities of the micro-partitions in the Shared-Processor Pool plus the Reserved Pool Capacity.

Using the information in Table 8-2 as an example and a Reserved Pool Capacity of 1.5, it is easy to calculate the Entitled Pool Capacity.

Table 8-2 Entitled capacities for micro-partitions in a Shared-Processor Pool

Micro-partitions in a Shared-Processor Pool	Entitled capacity for micro-partitions
Micro-partition 0	0.5
Micro-partition 1	1.75
Micro-partition 2	0.25
Micro-partition 3	0.25
Micro-partition 4	1.25
Micro-partition 5	0.50

The sum of the entitled capacities for the micro-partitions in this Shared-Processor Pool is 4.50, which gives us the following calculation:

$$EntitledPoolCapacity = 4.50 + 1.50$$

This gives the following result:

$$EntitledPoolCapacity = 6.0$$

8.2.2 The default Shared-Processor Pool (SPP₀)

The default Shared-Processor Pool (SPP₀) is automatically activated by the system and is always present. Its Maximum Pool Capacity is set to the capacity of the Physical Shared-Processor Pool. For SPP₀, the Reserved Pool Capacity is always 0.

The default Shared-Processor Pool has the same attributes as a user-defined Shared-Processor Pool except that these attributes are not directly under the control of the system administrator; they have fixed values.

Table 8-3 shows the attributes for the default Shared-Processor Pool.

Table 8-3 Attribute values for the default Shared-Processor Pool (SPP₀)

SPP ₀ attribute	Description	Value
Shared-Processor Pool ID	Default Shared-Processor Pool identifier	0
Maximum Pool Capacity	The maximum allowed capacity - the upper capacity boundary for the Shared-Processor Pool. For SPP ₀ , this value cannot be changed.	The value is equal to the number of active physical processors in the Physical Shared-Processor Pool (all capacity in the Physical Shared-Processor Pool). The number of processors can vary as physical processors enter or leave the Physical Shared-Processor Pool through dynamic or other partition activity.
Reserved Pool Capacity	Reserved processor capacity for this Shared-Processor Pool. For SPP ₀ , this value cannot be changed.	0
Entitled Pool Capacity	Sum of the capacity entitlements of the micro-partitions in SPP ₀ plus the Reserved Pool Capacity (which is always zero in SPP ₀).	Sum (total) of the entitled capacities of the micro-partitions in SPP ₀ .

The maximum capacity of SPP₀ can change indirectly through system administrator action such as powering on a dedicated-processor partition, or dynamically moving physical processors in or out of the Physical Shared-Processor Pool.

8.2.3 Server load and capacity examples

The example in Figure 8-1 shows a Shared-Processor Pool under load; some micro-partitions require more processing capacity than their entitled capacity.

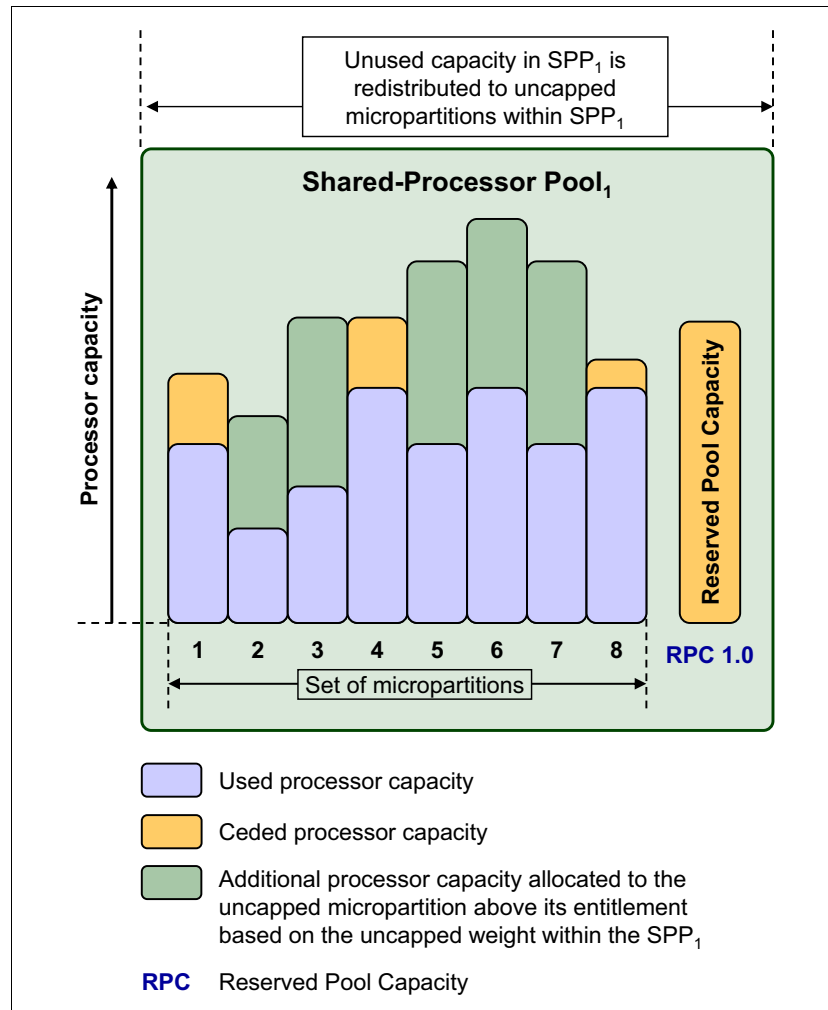


Figure 8-1 Redistribution of ceded capacity within Shared-Processor Pool

Each micro-partition in Shared-Processor Pool₁ (SPP₁) shown in Figure 8-1 is guaranteed to receive its processor entitlement if it is required. However, some micro-partitions in SPP₁ have not used their entire entitlement and so will cede the capacity. The ceded capacity is redistributed to the other uncapped micro-partitions within SPP₁ on an uncapped weighted basis.

However, the ceded capacity might not be enough to satisfy the demand, and so additional capacity can be sourced from the Reserve Pool Capacity of SPP_1 and distributed according to the uncapped weight of the requesting micro-partitions.

Figure 8-2 shows a case where Multiple Shared-Processor Pools have been created: SPP_1 , SPP_2 , and SPP_3 . Each Shared-Processor Pool has its own Reserved Pool Capacity (not shown), which it can distribute to its micro-partitions on an uncapped weighted basis. In addition, each Shared-Processor Pool can accumulate unused/ceded processor capacity from the under-utilized micro-partitions and again redistribute it accordingly.

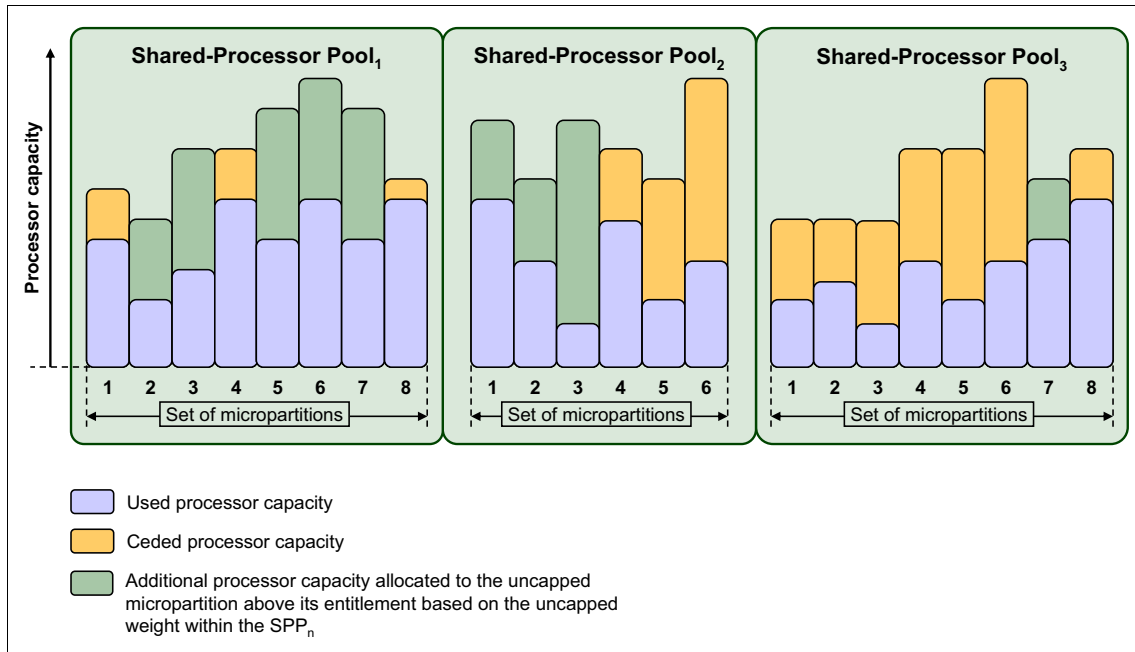


Figure 8-2 Example of Multiple Shared-Processor Pools

SPP₀: The default Shared-Processor Pool is not shown for the sake of clarity.

SPP₁ appears to be heavily loaded, as there is little unused capacity and several micro-partitions are receiving additional capacity. SPP₂ has a moderate loading, whereas SPP₃ is lightly loaded where most micro-partitions are ceding processor capacity.

Levels of processor capacity resolution

There are two levels of processor capacity resolution implemented by the POWER Hypervisor and Multiple Shared-Processor Pools:

- Level₀ The first level, Level₀, is the resolution of capacity within the same Shared-Processor Pool. Unused processor cycles from within a Shared-Processor Pool are harvested and then redistributed to any eligible micro-partition within the same Shared-Processor Pool.
- Level₁ When all Level₀ capacity has been resolved within the Multiple Shared-Processor Pools, the POWER Hypervisor harvests unused processor cycles and redistributes them to eligible micro-partitions regardless of the Multiple Shared-Processor Pools structure. This is the second level of processor capacity resolution.

Figure 8-3 depicts an example of a micro-partition moving from one Shared-Processor Pool to another.

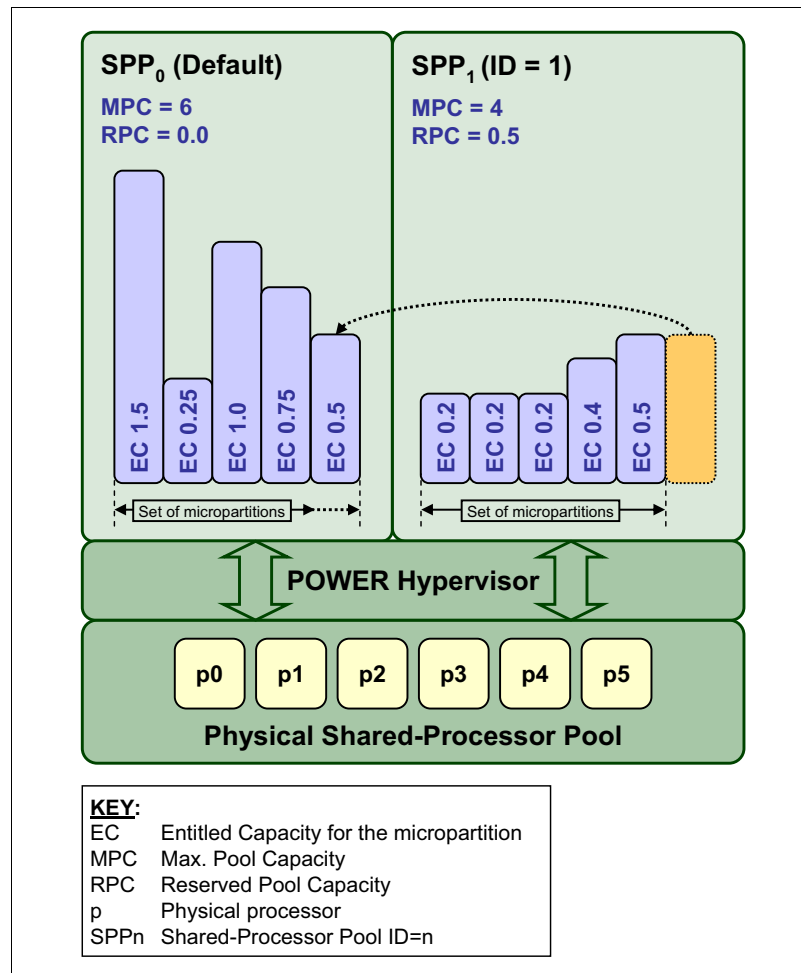


Figure 8-3 Example of micro-partition moving between Shared-Processor Pools

The movement of micro-partitions between Shared-Processor Pools is really a simple reassignment of the Shared-Processor Pool ID that a particular micro-partition is associated with. From the example in Figure 8-3, we can see that a micro-partition within Shared-Processor Pool₁ is reassigned to the default Shared-Processor Pool₀.

This movement reduces the Entitled Pool Capacity of Shared-Processor Pool₁ by 0.5 and correspondingly increases the Entitled Pool Capacity of the Shared-Processor Pool₀ by 0.5 as well. The Reserved Pool Capacity and Maximum Pool Capacity values are not affected.

Capacity: The Maximum Pool Capacity must be equal to or greater than the Entitled Pool Capacity in a Shared-Processor Pool. If the movement of a micro-partition to a target Shared-Processor Pool pushes the Entitled Pool Capacity past the Maximum Pool Capacity, then movement of the micro-partition will fail.

You can see the two levels of unused capacity redistribution implemented by the POWER Hypervisor in Figure 8-4.

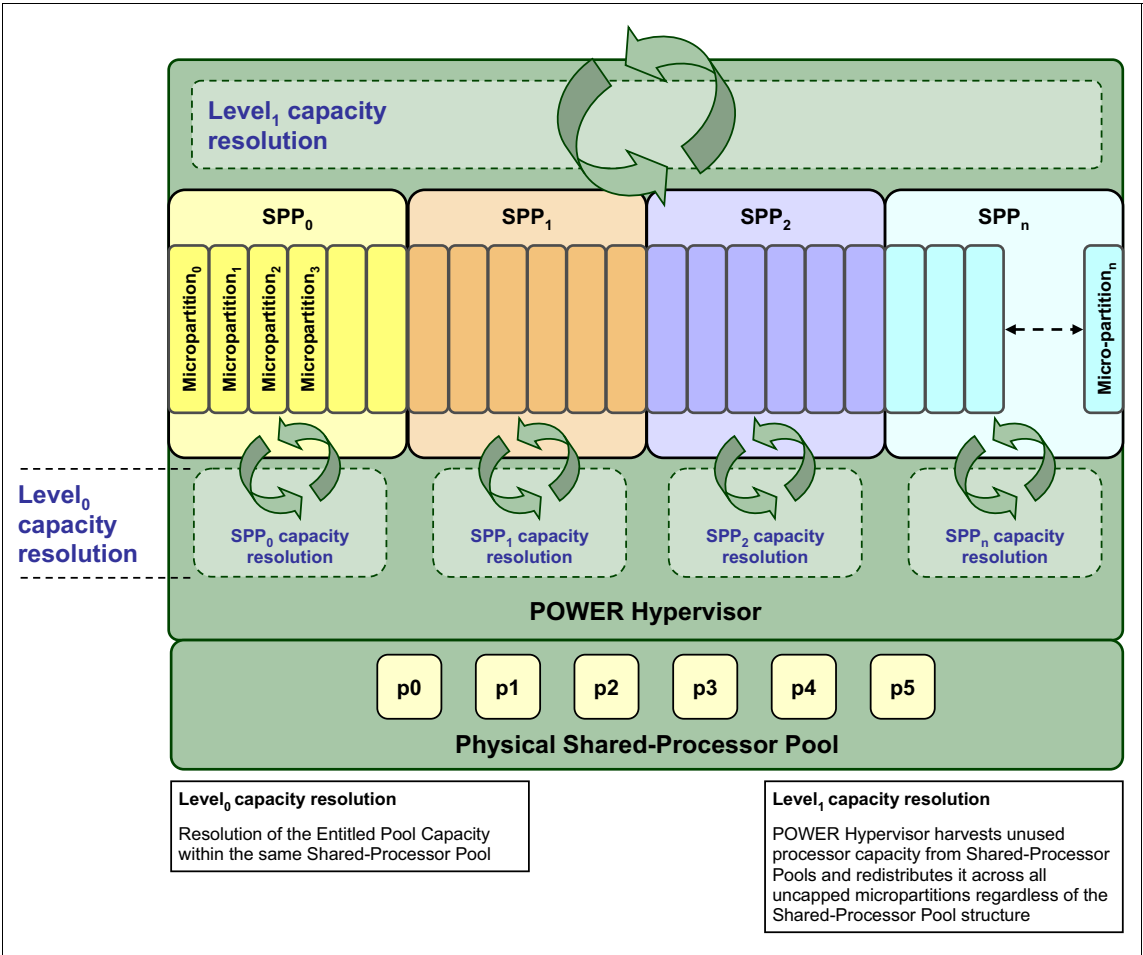


Figure 8-4 The two levels of unused capacity redistribution

8.2.4 Shared-Processor Pools scenarios

This section shows some Shared-Processor Pool example scenarios.

Figure 8-5 provides an example of how a Web-facing deployment maps onto a set of micro-partitions within a Shared-Processor Pool structure. There are three Web servers, two application servers, and a single database server.

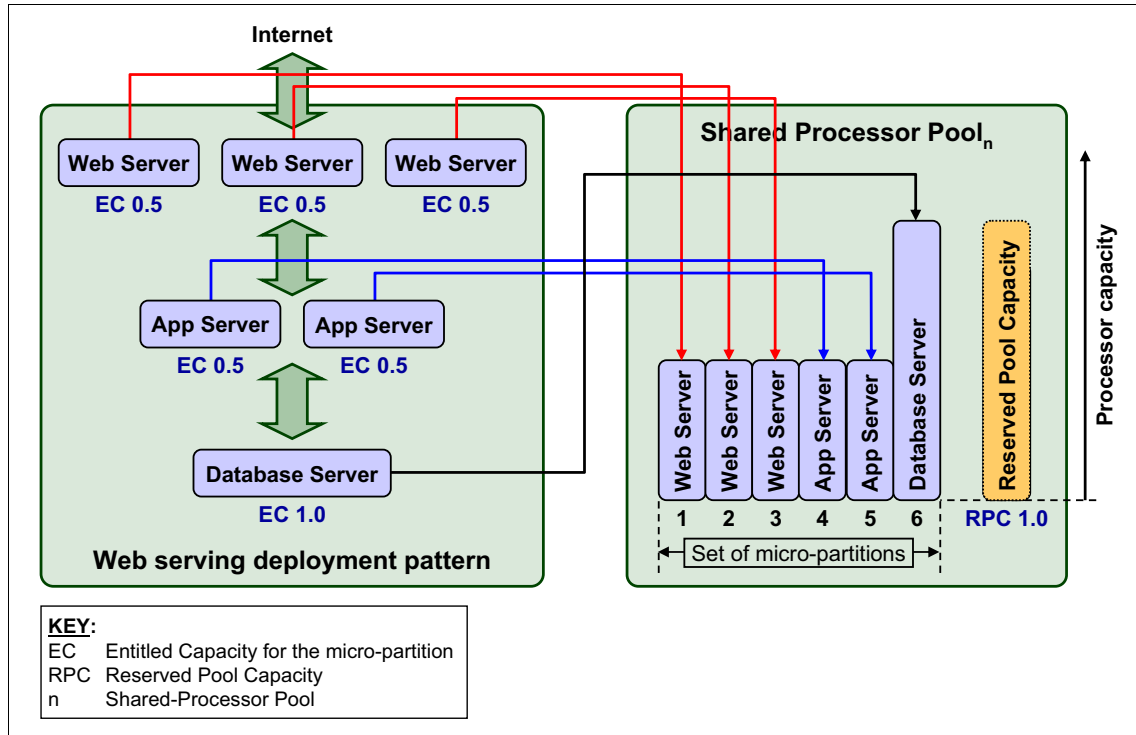


Figure 8-5 Example of a Web-facing deployment using Shared-Processor Pools

Each of the Web server micro-partitions and application server micro-partitions have an entitled capacity of 0.5 and the database server micro-partition has an entitled capacity of 1.0, making the total entitled capacity for this group of micro-partitions 3.5 processors. In addition to this, Shared-Processor Pool_n has a Reserved Pool Capacity of 1.0 which makes the Entitled Pool Capacity for Shared-Processor Pool_n 4.5 processors.

If you assume that all of the micro-partitions in Shared-Processor Pool_n are uncapped (and they have adequate virtual processors configured), then all micro-partitions will become eligible to receive extra processor capacity when required. The Reserved Pool Capacity of 1.0 ensures that there will always be some capacity to be allocated above the entitled capacity of the individual micro-partitions even if the micro-partitions in Shared-Processor Pool_n are under heavy load.

You can see this resolution of processor capacity within Shared-Processor Pool_n in Figure 8-6. The left of the diagram outlines the definition of Shared-Processor Pool_n and the micro-partitions that make it up. The example on the right of the diagram emphasizes the processor capacity resolution that takes place within Shared-Processor Pool_n during operation.

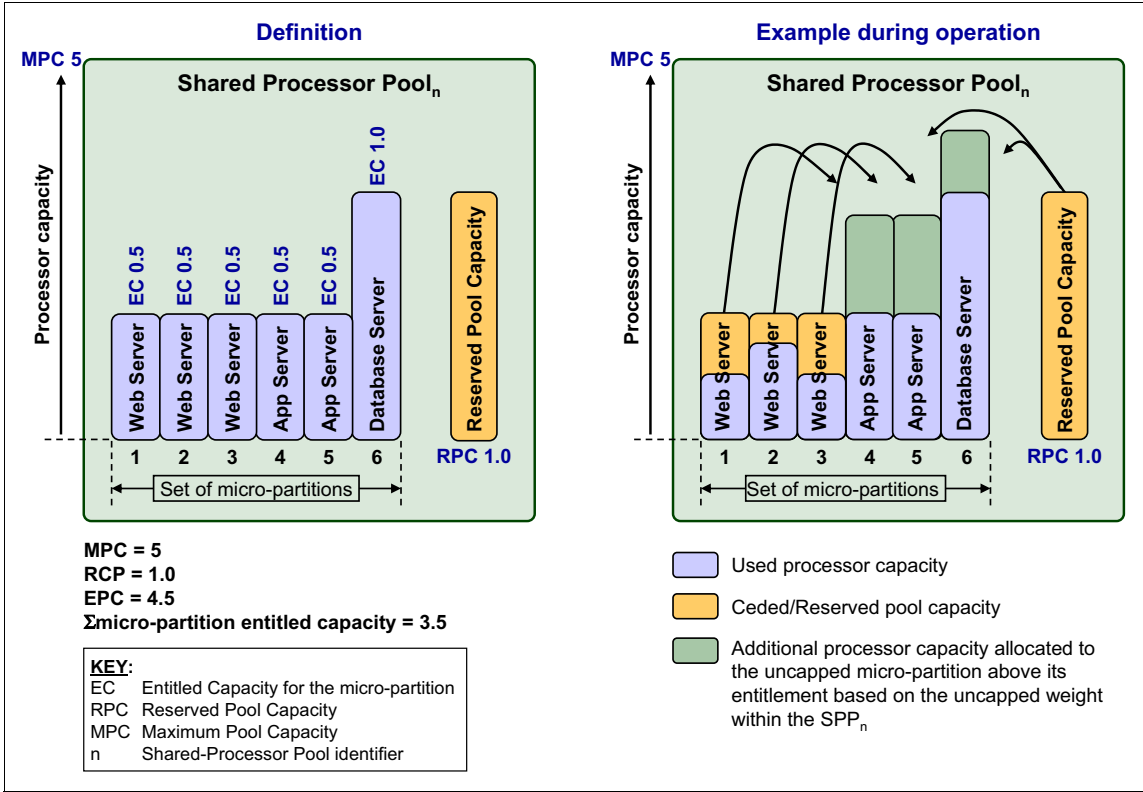


Figure 8-6 Web deployment using Shared-Processor Pools

In this example, during operation (one 10-ms POWER Hypervisor dispatch cycle) the Web servers are underutilized and cede processor capacity. However, the application servers and database server are heavily loaded and require far more processor cycles. These extra cycles are sourced from the Reserved Pool Capacity and the ceded capacity from the Web servers. This additional processor capacity is allocated to the application servers and database server using their uncapped weighting factor within Shared-Processor Pool_n (Level₀ capacity resolution).

You will notice that the Maximum Pool Capacity is 0.5 above the Entitled Pool Capacity of Shared-Processor Pool_n and so Level₁ capacity resolution can operate. This means that the uncapped micro-partitions within Shared-Processor Pool_n can also receive some additional processor capacity from Level₁ as long as the total capacity consumed is no greater than 5 processors (0.5 above the Entitled Pool Capacity).

The example shown in Figure 8-6 on page 128 outlines a functional deployment group, in this case a Web-facing deployment. Such a deployment group is likely to provide a specific service and is self-contained. This is particularly useful for providing controlled processor capacity to a specific business line (such as Sales or Manufacturing) and their functional applications.

There are other circumstances in which you might want to control the allocation of processor capacity and yet gain the advantages of capacity redistribution using the Multiple Shared-Processor Pools capabilities. In Figure 8-7, a set of micro-partitions are all database servers. You can see from the micro-partition definitions (left side of the diagram) the entitled capacity of each micro-partition, but there is no Reserved Pool Capacity.

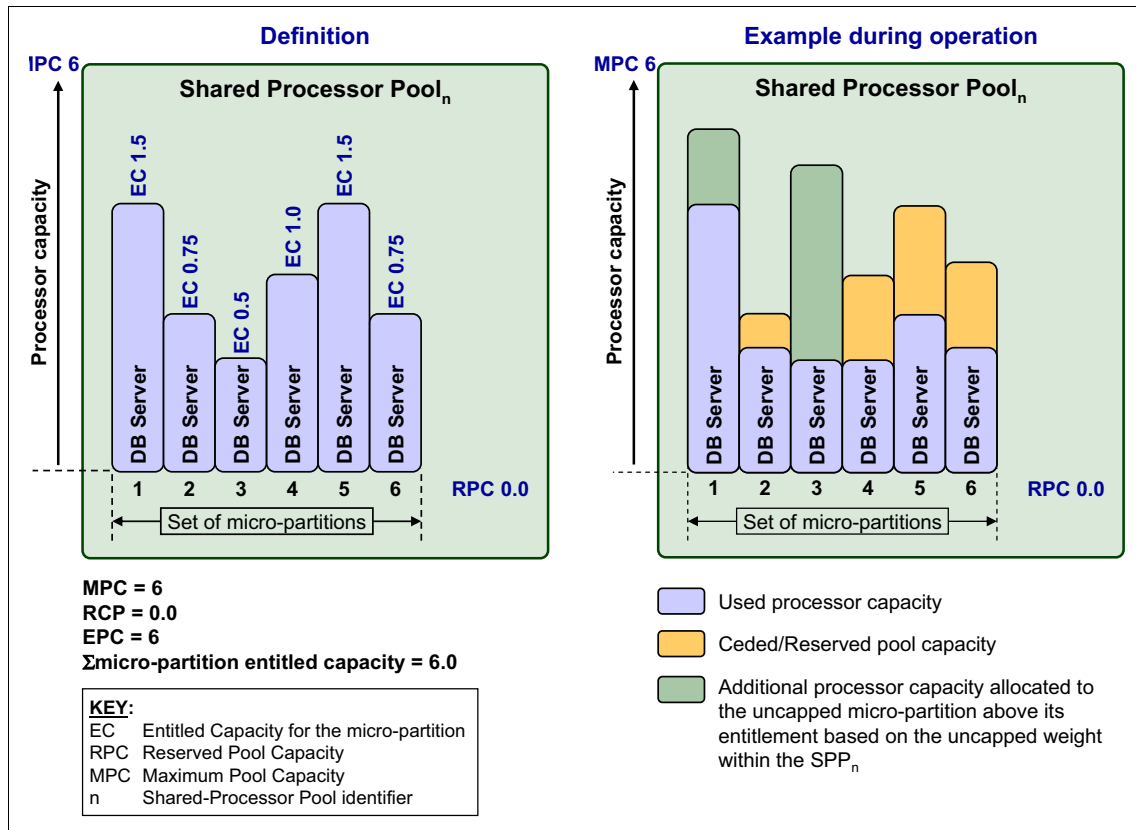


Figure 8-7 Capped Shared-Processor Pool offering database services

Also, Entitled Pool Capacity equals Maximum Pool Capacity ($EPC = MPC$). This essentially caps Shared-Processor Pool_n and prohibits the micro-partitions within Shared-Processor Pool_n from receiving any additional processor capacity from Level₁ capacity resolution.

Such an arrangement restricts the processor capacity for Shared-Processor Pool_n and therefore can restrict the software licensing liability, yet it provides the flexibility of processor capacity resolution within Shared-Processor Pool_n (Level₀ capacity resolution). This optimizes the use of any software licensing because the maximum amount of work is done for the investment in the software license.

In the example in Figure 8-7 on page 130, the Shared-Processor Pool_n configuration limits the processor capacity of 6 processors, which provides the opportunity to maximize the workload throughput for the corresponding software investment.

You can, of course, change this definition to include a Reserved Pool Capacity. This additional guaranteed capacity will be distributed to the micro-partitions within Shared-Processor Pool_n on an uncapped weighted basis (when a micro-partition requires the extra resources and has enough virtual processors to exploit it). For the example in Figure 8-7 on page 130, to accommodate an increase in Reserved Pool Capacity you will also have to increase the Maximum Pool Capacity. In addition, for the increase in the processor capacity for Shared-Processor Pool_n there will probably be an increase in the software licensing costs.

If the Maximum Pool Capacity is increased further so that it is greater than the Entitled Pool Capacity, then uncapped micro-partitions within Shared-Processor Pool_n can become eligible for Level₁ capacity resolution, additional processor capacity from elsewhere in the system. This can mean that any software for Shared-Processor Pool_n can likely be licensed for the Maximum Pool Capacity whether or not the micro-partitions in Shared-Processor Pool_n actually receive additional cycles above the Entitled Pool Capacity.

Figure 8-8 gives a simple example of a system with Multiple Shared-Processor Pools. One of the three Shared-Processor Pools is Shared-Processor Pool₀, the default Shared-Processor Pool.

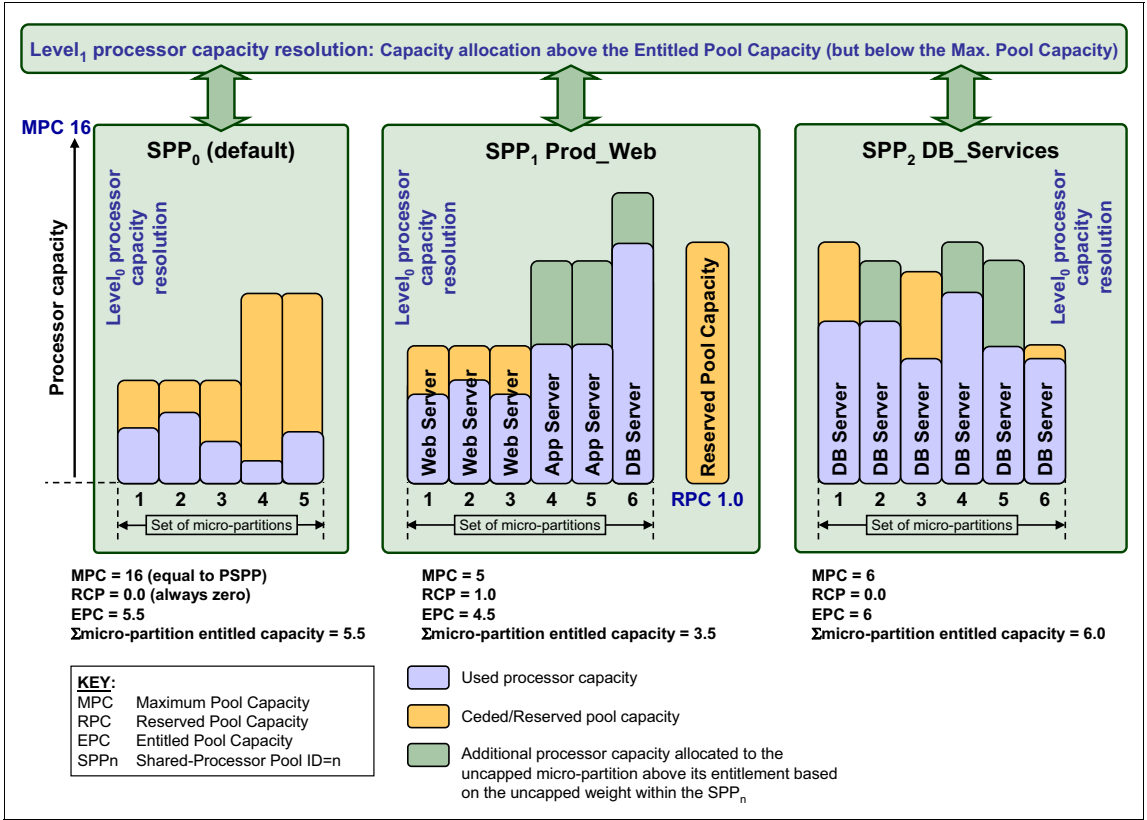


Figure 8-8 Example of a system with Multiple Shared-Processor Pools

As you can see, the Maximum Pool Capacity and the Entitled Pool Capacity for the DB_Services Shared-Processor Pool (Shared-Processor Pool₂) are equal, (MPC=EPC=6). Thus, DB_Services is effectively capped and cannot receive any additional processor capacity beyond that guaranteed to it by its Entitled Pool Capacity. Consequently, it is only affected by Level₀ capacity resolution.

However, the Prod_Web Shared-Processor Pool (SPP₁) has a Maximum Pool Capacity greater than its Entitled Pool Capacity. So the micro-partitions within Prod_Web can be allocated additional capacity using Level₁ capacity resolution.

Shared-Processor Pool₀ (default) and SPP₁ both participate in Level₁ capacity resolution under certain circumstances. As the attributes of SPP₀ are set to default values and cannot be altered by the systems administrator, it is always capable of consuming all the processor capacity in the physical shared-processor pool. Of course, this is assuming that at least one micro-partition in SPP₀ is uncapped and there are enough virtual processors to utilize the additional capacity.

8.3 Software license in a virtualized environment

In the past, a new processor might have provided a throughput gain with a speed increase over an existing generation. Nowadays, a new processor can improve throughput using other methods, such as these:

- ▶ Number of cores per physical chip
- ▶ Number of threads each core can simultaneously dispatch
- ▶ Size and speed of cache
- ▶ Specialist processing units, for example, decimal floating point in POWER6

The traditional measure for software licensing, the number and speed of processors, is no longer a direct comparison of chip performance. In addition to this vendors are introducing virtualization technologies, allowing the definition of logical servers that use fractions of processing power.

As the virtualization capabilities of Power Systems technology evolves, so must the software licensing models to support and take best advantage of this. A number of software vendors, including IBM, are working towards new licensing methods to best take advantage of these technologies. By making use of features such as Multiple Shared-Processor Pool technology (MSPP) there is an opportunity for software vendors to provide new and innovative licensing terms, allowing us to be able to define pools of processors supporting different licensed software.

The following sections show the factors to be considered when planning the license model to be used. A licensing factors summary is presented at the end.

8.3.1 Licensing factors in a virtualized system

With the mainstream adoption of virtualization, more and more Independent Software Vendors (ISVs) are adapting their licensing to accommodate the new virtualization technologies. A number of different models exist, varying with the ISVs. When calculating the cost of licensing and evaluating which virtualization technology to use, consider the following factors:

- ▶ ISV recognition of virtualization technology and capacity capping method
- ▶ ISV sub-capacity licensing available for selected software products
- ▶ ISV method for monitoring and management of sub-capacity licensing
- ▶ ISV flexibility as license requirements change

Cost of software licenses

A careful consideration of the licensing factors in advance can help reduce the overall cost in providing business applications. Traditional software licensing has been based on a fixed machine with a fixed amount of resources. With these new PowerVM technologies, there are a number of challenges to this model:

- ▶ It is possible to migrate partitions between different physical machines (with different speeds and numbers of total processors activated).
- ▶ Consider a number of partitions which, at different times, are all using four processors. However, these can all now be grouped using Multiple Shared-Processor Pool technology, which will cap the overall CPU always at four CPUs in total.

When the ISV support for these technologies is in place, it is anticipated that it will be possible to increase the utilization within a fixed cost of software licenses.

Active processors and hardware boundaries

The upper boundary for licensing is always the quantity of active processors in the physical system (assigned and unassigned), because only active processors can be real engines for software.

Above the physical system level, on Power Systems servers, partitions can be defined. Most software vendors consider each partition as a standalone server and, depending on whether it is using dedicated processors or micro-partitioning, will license software per partition.

The quantity of processors for a certain partition can vary over time, for example, with dynamic partition operations, but the overall licenses must equal or exceed the total number of processors used by the software at any point in time. If you are using uncapped micro-partitions, then the licensing must take into account the fact that the partition can use extra processor cycles above the initial capacity entitlement.

8.3.2 Capacity capping

There are two kinds of models for licensing software:

- ▶ Pre-pay license based on server capacity or number of users.
- ▶ Post-pay license based on auditing and accounting for actual capacity used.

With most software vendors offering the pre-pay method, the question most vendors will ask will be about how much capacity a partition can use. With this in mind, the following sections illustrate how to calculate the amount of processing power a partition can use.

Dedicated or dedicated donating partitions

In a partition with dedicated processors, the initial licensing needs to be based on the number of processors assigned to the partition at activation. Depending on the partition profile maximums, if there are additional active processors or Capacity Upgrade on Demand processors available in the system, these can be added dynamically allowing operators to increase the quantity of processors.

Consider the number of software licenses before any additional processors are added, even temporarily, for example, with dynamic partition operations. Clients need to note that some ISVs can require licenses for the maximum possible number of processors for each of the partitions where the software is installed (the maximum quantity of processors in the partition profile).

The sharing of idle processor cycles from running dedicated processor partitions will not change the licensing considerations.

Capacity capping of micro-partitions

A number of factors must be considered when calculating the capacity of micro-partitions. To allow the POWER Hypervisor to create micro-partitions the physical processors are presented to the operating system as virtual processors. As micro-partitions are allocated processing time by the POWER Hypervisor, these virtual processors are dispatched on physical processors on a time-share basis.

With each logical processor mapping to a physical processor, the maximum capacity an uncapped micro-partition can use is the number of available virtual processors, with the following assumptions:

- ▶ This capacity does not exceed the number of active processors in the physical system.
- ▶ This capacity does not exceed the available capacity in the Shared-Processor Pool.

The following sections discuss the different configurations possible and the licensing implications of each.

Capped micro-partition

For a micro-partition, the desired entitled capacity is a guaranteed capacity of computing power that a partition is given upon activation. For a capped micro-partition, the entitled capacity is also the maximum processing power the partition can use,

Using dynamic LPAR operations, you can vary this between the maximum and minimum values in the profile.

Uncapped micro-partition without MSPP technology

The entitled capacity given to an uncapped micro-partition is not necessarily a limit on the processing power. An uncapped micro-partition can use more than the entitled capacity if there are some available resources within the system.

In this case, on a Power Systems server using the Shared-Processor Pool or using only the default Shared-Processor Pool on POWER6 or later, the limiting factor for uncapped micro-partition is the number of virtual processors. The micro-partition can use up to the number of physical processors in the Shared-Processor Pool, because each virtual processor is dispatched to a physical processor.

With a single pool, the total resources available in the Shared-Processor Pool will be equal to the activated processors in the machine minus any dedicated (non-donating) partitions. This assumes that at a point in time all other partitions will be completely idle.

The total licensing liability for an uncapped partition without MSPP technology will be either the number of virtual processors or the number of processors in the default Shared-Processor Pool, whichever is smallest.

Uncapped micro-partition with MSPP technology

Similarly, the entitled capacity for an uncapped micro-partition is not necessarily a limit on the processing power. An uncapped micro-partition can use more than the entitled capacity if there are some available resources within the system.

For POWER6 (or later) servers using Multiple Shared-Processor Pool technology, it is possible to group micro-partitions together and place a limit on the overall group maximum processing units. After defining a Shared-Processor Pool group, operators can group specific micro-partitions together that are running the same software (software licensing terms permitting) allowing a pool of capacity that can then be shared among a number of different micro-partitions.

However, overall, the total capacity used at any point in time will never exceed the pool maximum.

8.3.3 System with Capacity Upgrade on Demand processors

Processors in the Capacity Upgrade on Demand (CUoD) pool do not count for licensing purposes until the following events happen:

- ▶ They become temporarily or permanently active and are assigned to partitions.
- ▶ They become temporarily or permanently active in systems with PowerVM technology and can be used by micro-partitions.

Clients can provision licenses of selected software for temporary or permanent use on their systems. Such licenses can be used to align with the possible temporary or permanent use of CUoD processors in existing or new AIX, IBM i, or Linux partitions.

For more information about processors on demand On/Off, see Appendix C, “Capacity on Demand” on page 683

8.3.4 Summary of licensing factors

Depending on the licensing model supported by the software vendor, it is possible to work out licensing costs based on these factors:

- ▶ Capped versus uncapped micro-partitions
- ▶ Number of virtual processors
- ▶ Unused processing cycles available in the machine, from dedicated donating partitions and other micro-partitions
- ▶ Multiple Shared-Processor Pool maximum
- ▶ Active physical processors in the system

An example of the license boundaries is illustrated in Figure 8-9.

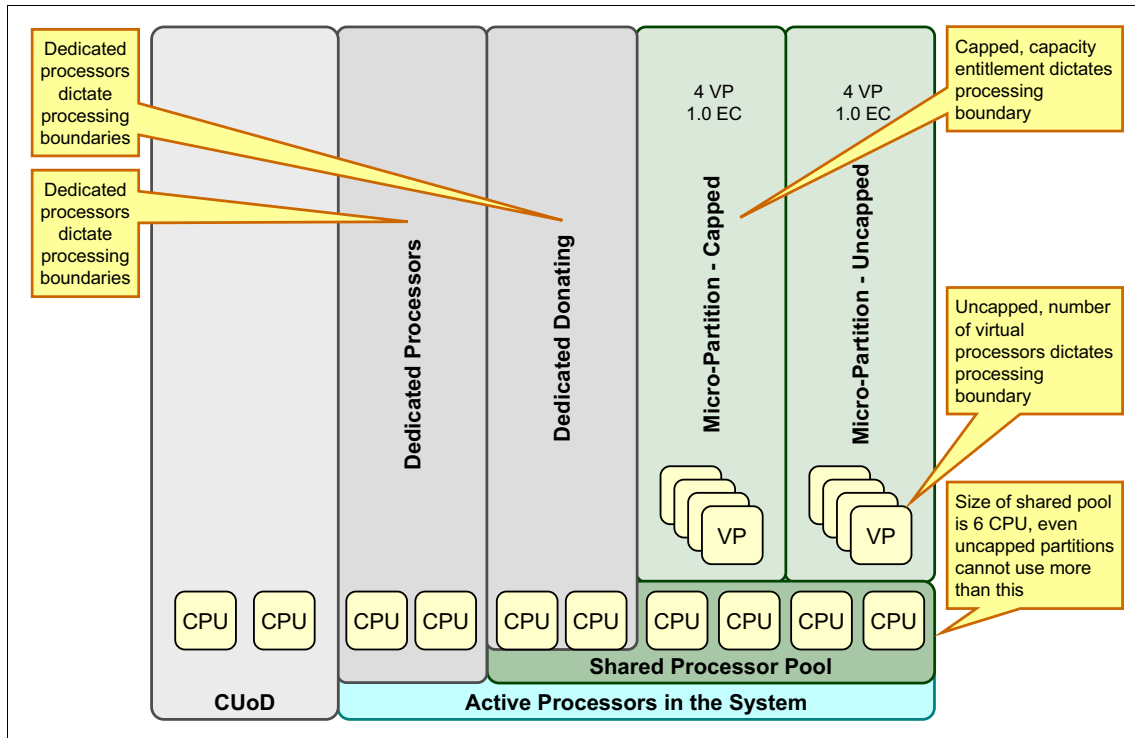


Figure 8-9 License boundaries with different processor and pool modes

8.3.5 IBM software

With the widespread introduction of multi-core chips, IBM software is moving to a licensing model based on a Processor Value Unit.

The Processor Value Units will allow the licensing of software to reflect the relative processor performance and allow the licensing of sub-capacity units.

For non-IBM software, contact your independent software vendor sales representative. For more information about IBM software licensing, see the links in 8.3, "Software license in a virtualized environment" on page 133.

Selected IBM software programs eligible under IBM Passport Advantage® and licensed on a per-processor basis can qualify for sub-capacity terms, so licenses are required only for those partitions where the programs are installed. To be eligible for sub-capacity terms, the client must agree to the terms of the IBM International Passport Advantage Agreement Attachment for sub-capacity Terms.

Capacity Upgrade on Demand

Only selected IBM software for the Power Systems offerings is eligible for on demand licensing. When planning for software charges on a per-processor basis for the systems, the client must also differentiate between these licenses:

Initial licensing	The client calculates the initial license entitlements, based on the licensing rules and the drivers for licensing. The client purchases processor licenses based on the planned needs. The client can also purchase temporary On/Off licenses of selected Power System related software.
Additional licensing	The client checks the actual usage of software licenses or future planned needs and calculates the additional license entitlements (temporary On/Off licenses also) based on the licensing rules and the drivers for licensing.
On demand licensing	The client contacts IBM or a Business Partner for the submission of a Passport Advantage Program enrollment. The client follows the procedures of the licensing method (sub-capacity licensing for selected IBM Passport Advantage eligible programs) including any auditing requirements and is billed for the capacity used.

Sub-capacity licensing for IBM software

IBM already offers some software under sub-capacity licenses. For more information about the terms, see this website:

<http://www-306.ibm.com/software/lotus/passportadvantage/licensing.html>

An implementation similar to the following one might be possible using sub-capacity licensing.

Consider a non-partitioned system with a fixed number of cores running an MQ and IBM DB2® workload. The licensing calculation might be similar to that in Figure 8-10.

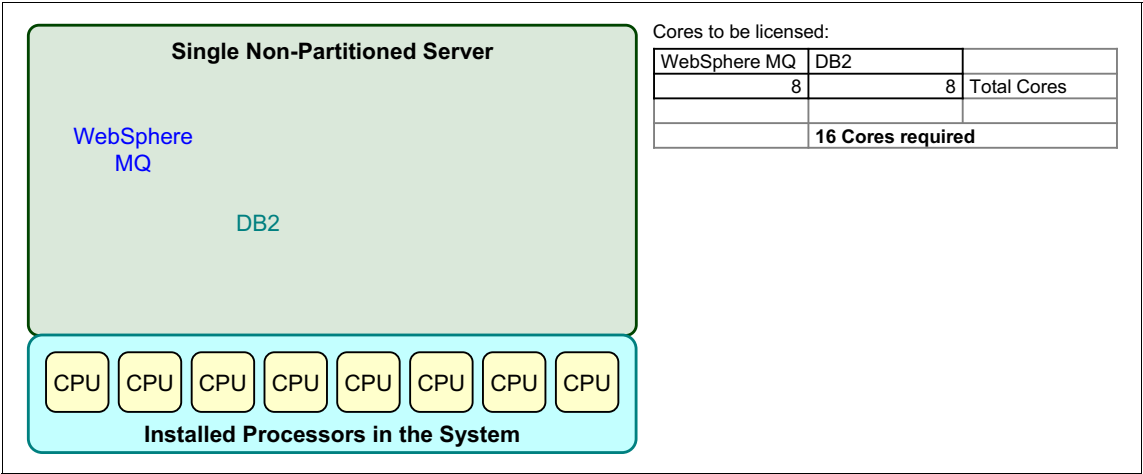


Figure 8-10 Licensing requirements for a non-partitioned server

Using Power System virtualization technology, it is possible to take advantage of micro-partitioning to reduce the licensing costs in a manner similar to that in Figure 8-11.

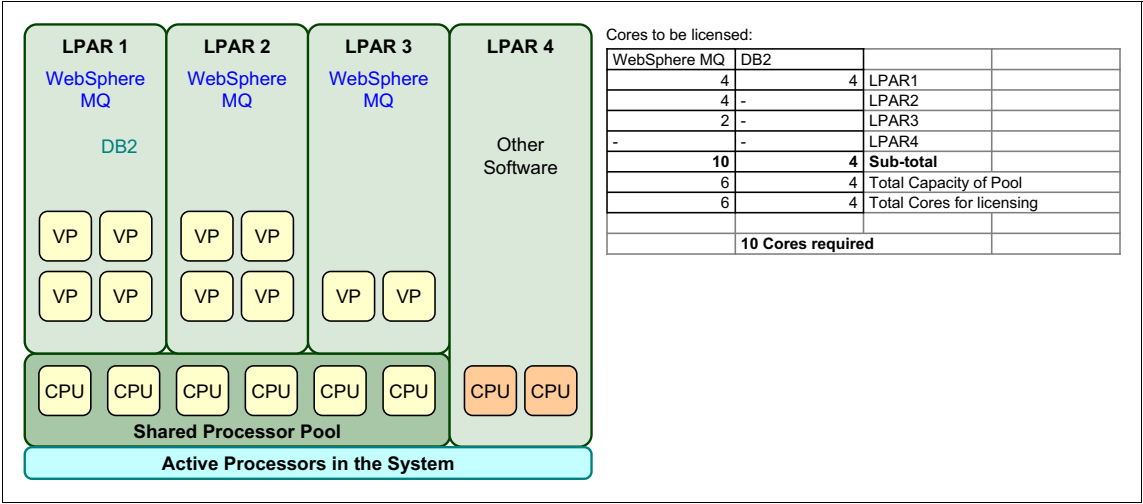


Figure 8-11 Licensing requirements in a micro-partitioned server

For additional information about terms and conditions for sub-capacity licensing of selected IBM software for your geography, contact your IBM representative or visit this website:

<http://www.ibm.com/software/passportadvantage>

8.3.6 Software licensing on IBM i

Starting with IBM i 7.1, it is possible to use workload groups to limit the processing capacity of a workload to a subset of processor cores in a partition. This capability requires the workload groups PTFs.

Consequently, workload groups can be used to reduce license costs for a processor usage type licensed program. To do that:

- ▶ Create a workload group with a maximum processor core limit that is less than the number of processor cores configured for the partition.
- ▶ Add the licensed program to the newly created workload group.
- ▶ Identify the workloads associated with the licensed program and associate the workloads with the newly created workload group.

The licensed program owner need to accept the reduced processor core capacity.

8.3.7 Linux operating system licensing

License terms and conditions of Linux operating system distributions are provided by the Linux distributor, but all base Linux operating systems are licensed under the GPL. Distributor pricing for Linux includes media, packaging/shipping, and documentation costs, and they can offer additional programs under other licenses as well as bundled service and support.

Clients or authorized business partners are responsible for the installation of the Linux operating system, with orders handled pursuant to license agreements between the client and the Linux distributor.

Clients need to consider the quantity of virtual processors in micro-partitions for scalability and licensing purposes (uncapped partitions) when installing Linux in a virtualized Power System server.

Each Linux distributor sets its own pricing method for their distribution, service, and support. Consult the distributor's website for information, or see these:

<http://www.novell.com/products/server/>
<https://www.redhat.com/>



Memory virtualization planning

This chapter describes the points that you need to plan for and verify before implementing Active Memory Sharing and Active Memory Deduplication in your environment.

It covers the following topics:

- ▶ Active Memory Sharing planning
- ▶ Active Memory Deduplication planning

9.1 Active Memory Sharing planning

This section describes the following topics related to planning for your Active Memory Sharing (AMS) environment:

- ▶ “Active Memory Sharing prerequisites” here provides a quick overview of all the prerequisites you need to deploy your Active Memory Sharing environment.
- ▶ “Deployment considerations” on page 145 discusses some basic guidelines to identify and choose your candidate workloads for Active Memory Sharing.
- ▶ “Sizing Active Memory Sharing” on page 151 describes tools and procedures to size your Active Memory Sharing environment.

9.1.1 Active Memory Sharing prerequisites

Table 9-1 provides the minimum levels required for Active Memory Sharing.

Table 9-1 Active Memory Sharing requirements

Component	Minimum level POWER 6	Minimum level POWER 7	Comments
Hardware	POWER6 processor-based server	POWER7 processor-based server	This hardware level is required because it provides the mechanism to enable AMS
I/O devices	Virtual only	Virtual only	All I/O devices must be virtualized by the Virtual I/O Server
Managed system firmware	340_075	710_043	Verify this on the ASMI home screen, or using the HMC
PowerVM	Enterprise edition	Enterprise edition	A hardware feature code
Management console	HMC 7.3.4 Service Pack 3	HMC 7.7.2 Service Pack 1	Check this on HMC/IVM home screen. Use Updates link to check version.
Virtual I/O Server	2.1.0.1-Fix Pack 21	2.1.3.10-Fix Pack 23	Issue ioslevel at the Virtual I/O Server shell

Component	Minimum level POWER 6	Minimum level POWER 7	Comments
AIX	6.1.3.0 Technology Level 3	6.1.3.0 Technology Level 4	Issue <code>oslevel -s</code> at the AIX shell
IBM i	IBM i 6.1.1 plus latest cumulative PTF package	IBM i 6.1.1 plus latest cumulative PTF package	Issue DSPPTF at the IBM i command line.
Linux	RHEL 6 or SLES 11	RHEL 6 or SLES 11	► Issue <code>cat /etc/SuSE-release</code> in SuSE SLES Linux shell ► Issue <code>cat /etc/redhat-release</code> in RedHat Linux shell

Attention: Non-supported operating systems will fail to boot in a logical partition configured for shared memory.

9.1.2 Deployment considerations

With Active Memory Sharing, you can optimize memory utilization through consolidation of workloads and achieve increased overall memory utilization. In the following sections, we describe the concepts and basic rules to identify those workloads and how to deploy an Active Memory Sharing environment.

Overcommitment

In an Active Memory Sharing environment, you can configure more logical memory than the available physical memory configured in the shared memory pool, resulting in an overcommitment scenario.

Each shared memory partition perceives itself as owning more of the memory, which results in total logical memory being oversubscribed.

Non-overcommit

The amount of real memory available in the shared pool is enough to cover the total amount of logical memory configured. Figure 9-1 shows partitions in which the logical memory does not exceed the shared memory pool size.

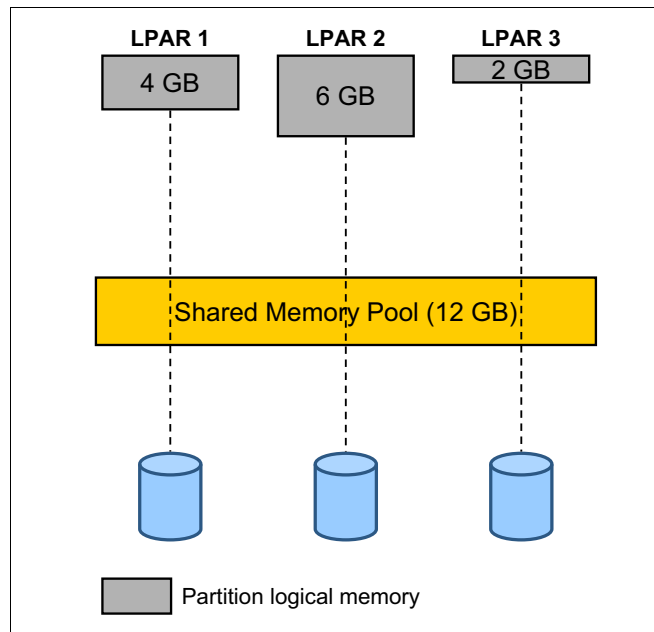


Figure 9-1 Non-overcommit

Logical overcommit

The logical memory in use at a given time is equal to the physical memory available in the shared memory pool. That is, the total logical configured memory can be higher than the physical memory; however, the working set never exceeds the physical memory. Figure 9-2 shows three logical partitions defined with 10 GB of logical memory each, and the sum of their memory usage never exceeds the shared memory pool size, which is 12 GB, across time.

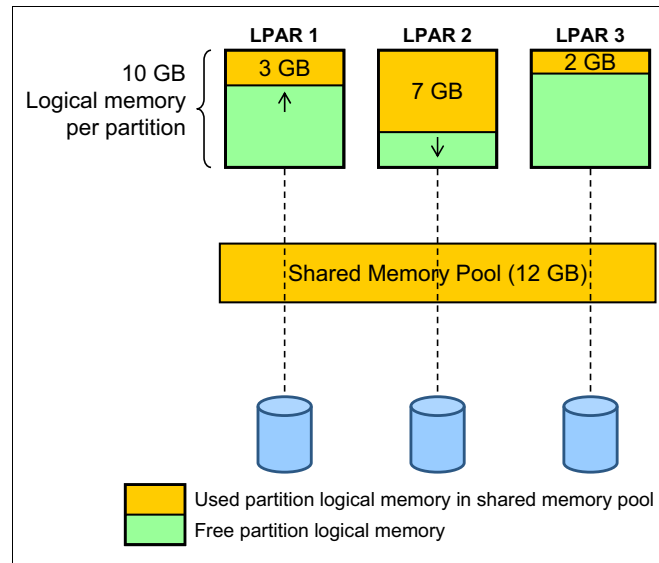


Figure 9-2 Logical overcommit

Physical overcommit The working set memory requirements can exceed the physical memory in the shared pool. Therefore, logical memory has to be backed by both the physical memory in the pool and by the paging devices. Figure 9-3 shows LPAR 3 as having 1 GB of its used memory on a paging device because the sum of the used memory is more than the size of the shared memory pool.

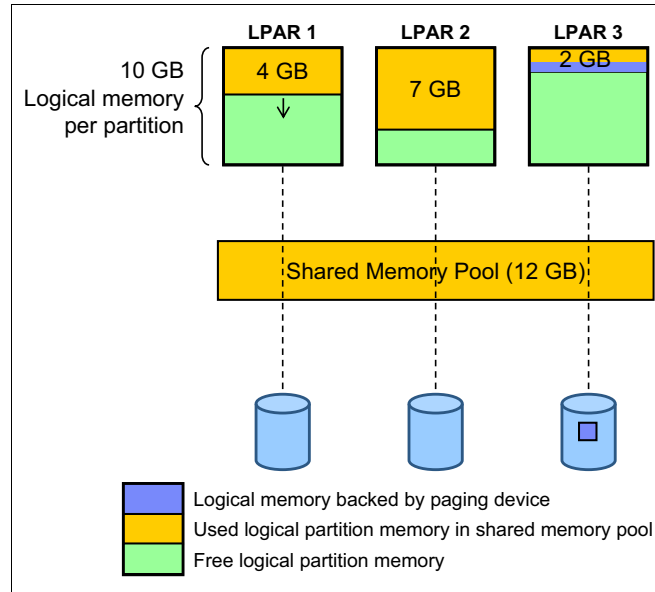


Figure 9-3 Physical overcommit

In the case of physical overcommitment, the hypervisor backs the excess logical memory using paging devices that are accessed through its paging Virtual I/O Server.

Workload selection

When selecting workloads to deploy using Active Memory Sharing, you first have to understand the workloads and the memory requirements of those workloads. It is important to monitor the workload peaks and valleys for a period of time (day/week/month) in dedicated memory mode to appropriately size the shared memory pools.

If you find deployed workloads that are not maximizing physical memory consumption, those workloads would be the prime candidates for Active Memory Sharing. In addition, there are some general principles to consider for workload selection.

As described in “Overcommitment” on page 146, there are multiple scenarios where memory can be overcommitted. Along with the workload selection, memory configuration decisions must be made. The following sections offer general rules to select what workloads fit which scenarios.

Logical overcommit

Active Memory Sharing logical overcommit is favored for workloads that have the following characteristics:

- ▶ Workloads that time multiplex. For example, in AM/PM scenarios, peaks and valleys of multiple workloads overlap leading to logical overcommit levels without consuming more than the physical memory available in the pool.
- ▶ Workloads that have low average memory residency requirements, where consolidating them would lead to logical overcommit.
- ▶ Workloads that do not have sustained loads, such as retail headquarters and university environments.
- ▶ Failover and backup partitions that are used for redundancy that require resources only when the primary server goes down with resources that do not have to be dedicated to redundant servers.
- ▶ Test and development environments.
- ▶ Private Cloud Computing environments.

Physical overcommit

Active Memory Sharing physical overcommit is favored for workloads that have the following characteristics:

- ▶ Workloads that currently run on the AIX operating system and use a lot of AIX file cache.
- ▶ Workloads that are less sensitive to I/O latency such as file servers, print servers, and network applications.
- ▶ Workloads that are inactive most of the time.
- ▶ Public Cloud Computing environments.

Dedicated memory partitions

Dedicated memory partitions are appropriate for the following workloads:

- ▶ Workloads that have high quality of service requirements.
- ▶ Workloads that have high sustained memory consumption due to sustained peak load.
- ▶ Workloads that mandate predictable performance.

- ▶ High Performance Computing workloads such as scientific computational intensive workloads that have sustained high CPU utilization and have high memory bandwidth requirements.

Consolidation factors

After workloads are selected for consolidation, the following factors must be considered:

- ▶ Logical to physical memory ratio suitable for the selected workloads.
- ▶ Shared memory weight determination for the workloads, assigning priority to your workloads.
- ▶ Based on the memory ratio, determine paging device configuration. And optimize the paging device accordingly.
- ▶ In a physical overcommit environment, determine whether aggressive loaning combined with the application load levels provide acceptable performance. The loaning level is set up on an operating system basis. Therefore, a mix of loaning levels can coexist on the same system.
- ▶ Optimize utilization through rebalancing resources. Physical memory might limit the number of shared processor partitions even though the system CPU and memory bandwidth utilization levels are low.

Paging device planning

Planning an Active Memory Sharing paging device is no different from planning for an operating system paging device.

When paging occurs, you want to perform it as fast as possible when high performance is a priority. Follow these guidelines to configure your paging devices for maximum performance and availability:

- ▶ Configure some level of disk redundancy, such as mirroring or RAID5.
- ▶ Use a small stripe size.
- ▶ Spread the I/O load across as much of the disk subsystem hardware as possible.
- ▶ Use a write cache on either the adapter or storage subsystem.
- ▶ Where possible, use physical volumes in preference over logical volumes.
- ▶ Size your storage hardware according to your performance needs.
- ▶ Ensure that PVIDs for paging devices for physical volumes set up by the HMC are cleared before use.

9.1.3 Sizing Active Memory Sharing

This section describes some models that can be used to size the requirements for Active Memory Sharing.

Virtual I/O Server resource sizing

The Virtual I/O Server should be configured as before for its standard disk and network I/O hosting for LPARs. The IBM Systems Workload Estimator tool can be used to help size the Virtual I/O Server server. For more information about this tool, see this website:

<http://www.ibm.com/systems/support/tools/estimator/index.html>

With Active Memory Sharing, the Virtual I/O Server takes a new role. If designated as a paging Virtual I/O Server, then the storage subsystem and paging rate must also be taken into consideration.

Table 9-2 provides a guide for estimating CPU entitlement per shared memory partition depending on estimated page rate and storage subsystem type.

Table 9-2 Estimated CPU entitlement requirements based on activity and storage

Paging rate	Storage type			
	Internal storage	Entry level storage	Mid range storage	High end storage
Light	0.005	0.010	0.020	0.020
Moderate	0.010	0.020	0.040	0.080
Heavy	0.020	0.040	0.080	0.160

Note: The heavier the paging rate, the more I/O operations are occurring within the Virtual I/O Server partition, thus resulting in an increased need for processor entitlement.

Figure 9-4 plots the information from Table 9-2 on page 151 to show the relationship between previous values.

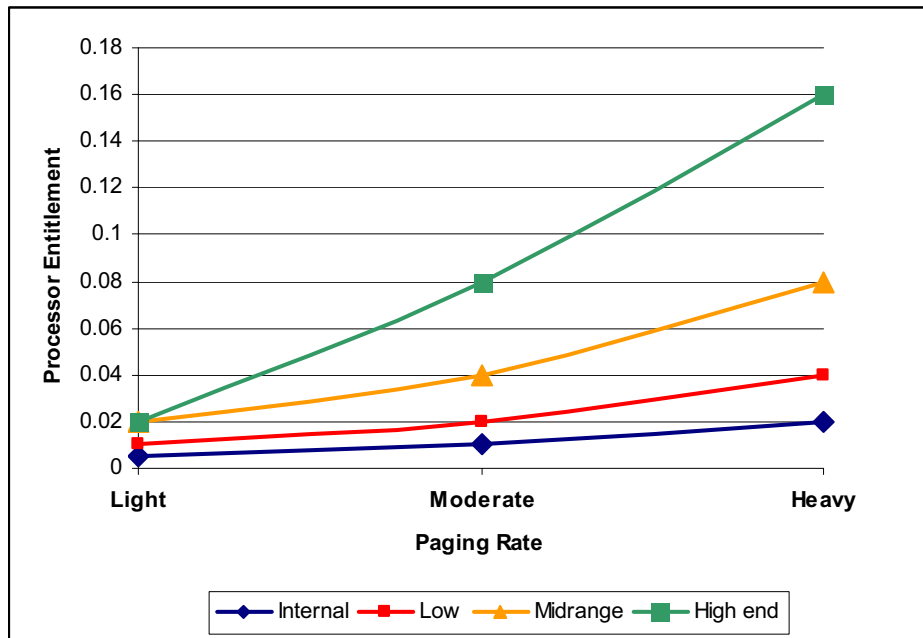


Figure 9-4 Est. additional CPU entitlement needed to support AMS paging device

As an example, if five shared memory partitions are being deployed, using a midrange storage solution for the shared memory pool paging devices, it would be recommended to use an additional $(5 * 0.04) = 0.2$ CPU entitlement for the Virtual I/O Server to ensure that enough computing resources are available to service five shared memory partitions simultaneously experiencing moderate hypervisor paging.

Important: You can deploy a Virtual I/O Server just to perform AMS paging and therefore separate paging I/O from other Virtual I/O Server functions, such as virtual disk and virtual network I/O in case of performance concerns. It is recommended to test a configuration before going into production.

Shared memory partition CPU sizing

Most of the tasks associated with Active Memory Sharing require the hypervisor to consume additional CPU cycles. In an oversubscription memory configuration, this increase in CPU utilization happens to be a function of:

- ▶ The access rate of the pages of physical memory.
- ▶ Physical to logical memory ratio
- ▶ Rate of disk access

If the memory is not overcommitted, the CPU utilization increase will be minimal due to the additional software layer of virtualization.

9.1.4 Usage examples

The goal of memory sharing is to optimize the usage of the memory pool by assigning the physical memory to the logical partitions that need it most at a specific point in time. This optimization can be either used to reduce global memory requirements of logical partitions or to allow logical partitions to increase their memory footprint during peak memory demand periods.

Multiple memory sharing scenarios are possible, depending on the overcommitment type of physical memory. The following sections discuss the different types of memory overcommitment and their advantages with respect to workload types.

Logical memory overcommitment

In a logical overcommitment scenario, the memory sizing of the shared memory partitions is made taking into account memory demands throughout an interval, such as a day, and making sure that the global requirement of physical memory never exceeds the physical memory in the pool.

In this configuration, it is possible to optimize existing physical memory on the system or to reduce the global memory needs.

Existing memory optimization

Consider a set of logical partitions that have been correctly sized and are using dedicated memory. The logical partition may be changed into shared logical partitions in a memory pool that contains the same amount of memory that the previously dedicated partitions occupied. The logical memory assigned to each shared memory partition is configured to be larger than the size in dedicated mode.

For example, four logical partitions with 10 GB of dedicated memory each can be configured to share a memory pool of 40 GB, each with 15 GB of logical memory assigned.

This new configuration does not change the global memory requirements, and every logical partition can have the same amount of physical memory it had before. However, memory allocation is highly improved since an unexpected memory demand due to unplanned peak of one logical partition can be satisfied by the shared pool. In deed unused memory pages from shared-memory partitions can be automatically assigned to the more demanding one automatically.

As long as the shared memory pool does not need to provide extra pages at the same moment to all logical partitions, hypervisor paging will be minimal.

If hypervisor paging activity increases too much, it is possible to add additional memory to the pool, and all shared memory partitions will take advantage of the increase in memory availability.

This differs from a dedicated memory configuration where the new memory would have to be statically assigned to only a few selected logical partitions.

Reduced memory needs

Good knowledge of physical memory requirements over time of multiple partitions allows you to configure systems with a reduced memory configuration.

For example, two logical partitions are known to require 8 GB each at peak time, but their concurrent requirement never exceeds 10 GB. The shared memory pool can be defined with 10 GB available and each logical partition is configured with 10 GB of logical memory. On a dedicated memory configuration, 16 GB of memory are required instead of the 10 GB required with the shared memory setup. This scenario is shown in Figure 9-5 with two AIX logical partitions.

The db_server logical partition starts a job that allocates 7 GB of logical memory while the web_server logical partition is kept idle. Partition db_server progressively increases the real memory usage based on how the job accesses the logical memory pages. The partition web_server decreases its real memory usage accordingly. The hypervisor asks AIX to loan memory and steals some memory pages.

When the db_server partition ends its job, another job on the web_server is started using 7 GB of logical memory. Following the new job memory accesses, the hypervisor starts to move memory pages from the db_server partition that is idle and assigns them to the web_server partition with the same loaning and stealing technique. See Figure 9-5.

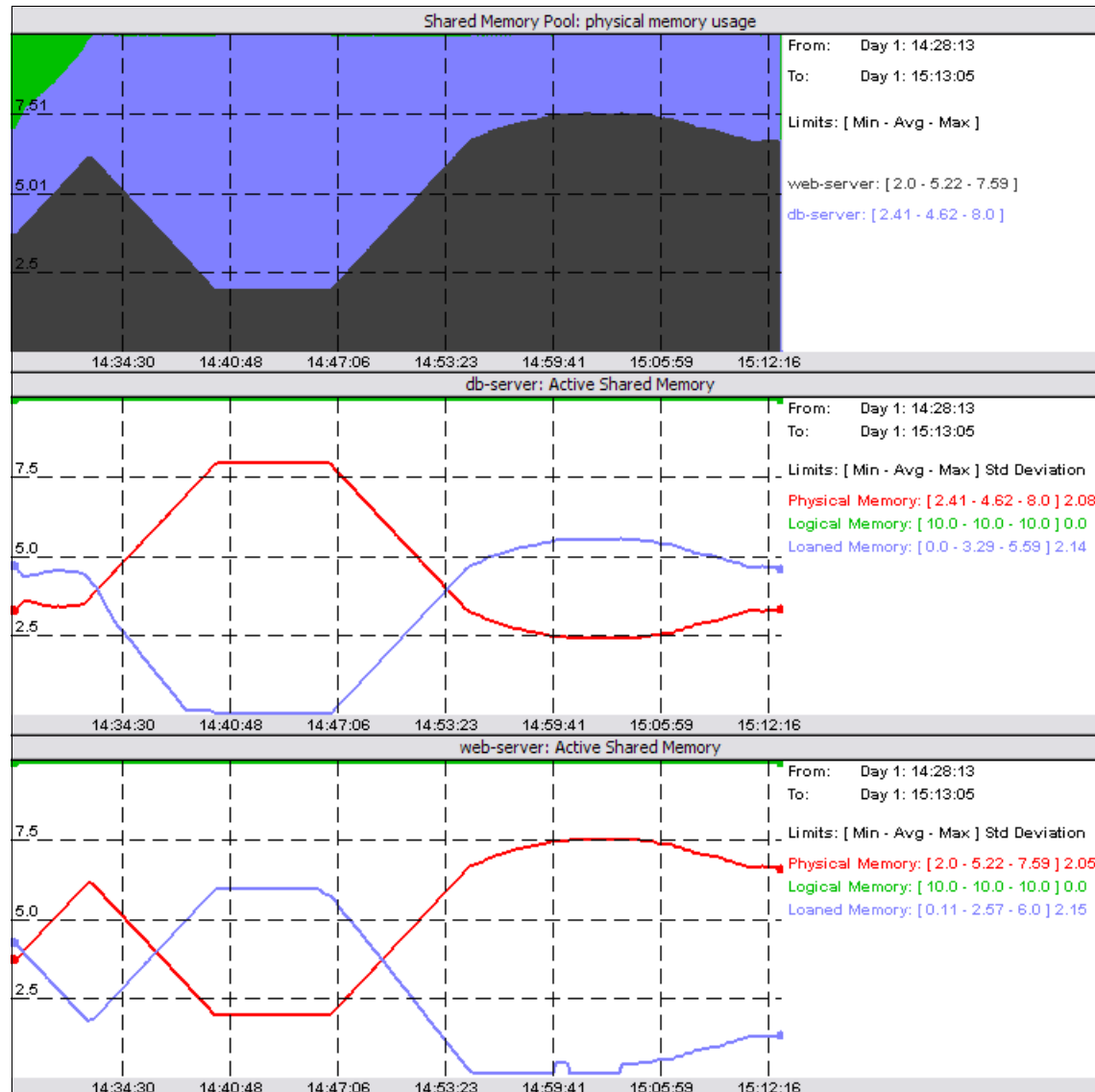


Figure 9-5 Logical overcommitment of memory example

The memory footprint required to host a set of shared memory partitions may be greatly reduced. Since shared memory partitions do not actively use all their memory at the same time, the hypervisor work is limited, and most of the stolen pages may be free pages and little I/O is required.

The hypervisor monitors memory demand on each shared memory partition and is able to meet the memory demand. When needed, page stealing is always started from the logical partitions that have lower memory activity in order to limit disk I/O operations.

The period in time when the hypervisor may have to perform high activity is during the transition of workloads from one logical partition to another. The first logical partition probably holds most of the memory even if it does not actively access it, while the second needs to allocate additional physical memory for its increased activity. During this short period, applications may monitor some increased memory access latency that is relieved as soon as the hypervisor finishes memory reallocation.

Logical overcommitment may be a good opportunity for workloads that have the following characteristics:

- ▶ They overlap peaks and valleys in memory usage. For example night and day activities or applications accessed by users in different time zones.
- ▶ They have low average memory residency requirements.
- ▶ They do not have sustained loads such as retail headquarters and university environments.
- ▶ Fail-over and backup logical partitions that are used for redundancy which require resources only when the primary server goes down. Resources do not have to be dedicated to a redundant server.

Physical memory overcommit

Physical overcommitment occurs when the sum of all logical memory that is actually being referenced at a point in time exceeds the physical memory in the shared memory pool. The hypervisor must then make frequent use of the paging devices to back up the active memory pages.

In this scenario, memory access times vary depending on whether logical pages are available in physical memory or on a paging device. The rate of hypervisor page faults determines application performance and throughput, but all logical partitions are allowed to work.

Not all workloads will be affected by memory latency, and overcommitment allows the creation of a larger number of logical partitions than with a dedicated memory configuration. There are scenarios where hypervisor paging is well suited for the configuration's needs.

Here are example configurations where physical overcommit may be appropriate:

- ▶ Workloads that use a lot of file cache. The operating system can control cache size according to hypervisor requirements in order to limit hypervisor paging.
- ▶ Workloads that are less sensitive to memory latency, such as file servers, print servers, and some network applications.
- ▶ Logical partitions that are inactive most of the time.

9.2 Active Memory Deduplication planning

This section presents the points that you need to plan for and verify before implementing Active Memory Deduplication in your environment.

It includes the following topics:

- ▶ Checking the requirements for Active Memory Deduplication
- ▶ Sizing your systems for Active Memory Deduplication

9.2.1 Checking the requirements for Active Memory Deduplication

Power Systems Active Memory Deduplication is intended to be used with IBM PowerVM Active Memory Sharing. Most of the prerequisites of Active Memory Deduplication are the same as the ones for Active Memory Sharing. Although Active Memory Sharing is supported on IBM POWER6 Systems, Active Memory Deduplication is currently supported only on IBM POWER7 Systems.

Table 9-3 lists the prerequisites for Active Memory Deduplication.

Table 9-3 Prerequisites for Active Memory Deduplication

Component	Minimum level	Comments
Hardware	POWER7 processor- based servers with firmware at level 7.4, or later	POWER7 is the minimum hardware level that supports Active Memory Deduplication.
I/O devices	Virtual only	All I/O devices must be virtualized by the Virtual I/O Server (VIOS).
Managed system firmware	Firmware level 7.4, or later	Validate through the hardware management interface.
PowerVM	PowerVM Enterprise Edition	You must have a hardware feature code or CoD-enabled feature.
Management console	Hardware Management Console (HMC) Version 7 Release 7.4.0, with mandatory eFix MH01235, or later	Check this version on the HMC home screen. At the time of writing this paper, the Integrated Virtualization Manager (IVM) does not support Active Memory Deduplication.
Virtual I/O Server	Virtual I/O Server (VIOS) Version 2.1.1.10 Fix Pack 21	Issue <code>ioslevel</code> at the Virtual I/O Server shell.
AIX	AIX Version 6 Technology Level 7, or later AIX version 7 Technology Level 1, or later	Issue <code>oslevel -s</code> at the AIX shell.
IBM i	IBM i 7.1.4 or later	Issue <code>DSPPTF</code> at the IBM i command line.
Linux	Novell SUSE Linux Enterprise Server 11 Service Pack 2 RedHat Enterprise Linux Version 6.2	► Issue <code>cat /etc/SuSE-release</code> in SuSE SLES Linux shell ► Issue <code>cat /etc/redhat-release</code> in RedHat Linux shell

Older versions of operating systems: Older versions of the supported operating systems down to the following levels can work with Active Memory Deduplication but without features such as statistics reporting and Active Memory Deduplication page hints:

- ▶ AIX: 6.1 TL4 or 7.1
- ▶ IBM i: IBM i 6.1.1 + latest cumulative PTF package
- ▶ Linux: SLES 11 and RHEL 6

The methods to check the system firmware and HMC firmware are beyond our discussion. You can find the information from the IBM websites.

Active Memory Sharing and Active Memory Deduplication support on IBM POWER systems: Although Active Memory Sharing is supported on POWER6 systems, Active Memory Deduplication is only supported on POWER7 systems.

IBM PowerVM edition

Active Memory Sharing and Active Memory Deduplication are supported only on the PowerVM Enterprise Edition. To check the capability of a managed system to use Active Memory Sharing and Active Memory Deduplication, you can use the GUI or CLI of the HMC. This section shows the procedure to check capability using all of these methods.

Checking system capabilities using the HMC GUI

To check the capability of a managed system using the HMC GUI, perform the following steps:

1. In the left pane of the Welcome window of the HMC, expand **Systems Management** → **Servers**. A list of managed servers is displayed under **Servers**.

2. In the Servers panel of the HMC, check the check box for the desired managed system. Select **Properties**, as shown in Figure 9-6.

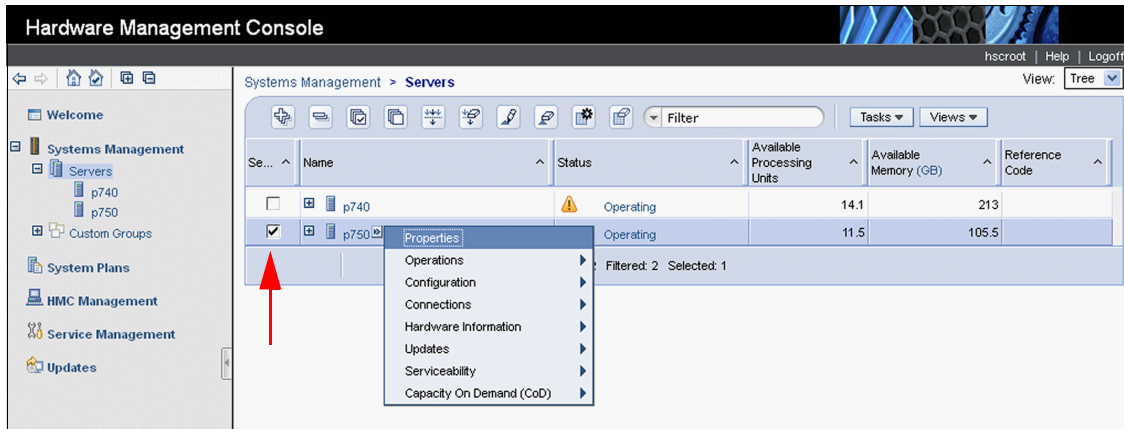


Figure 9-6 Selecting Properties for a managed system in the HMC

3. In the Properties window, select the **Capabilities** tab, as shown in Figure 9-7.

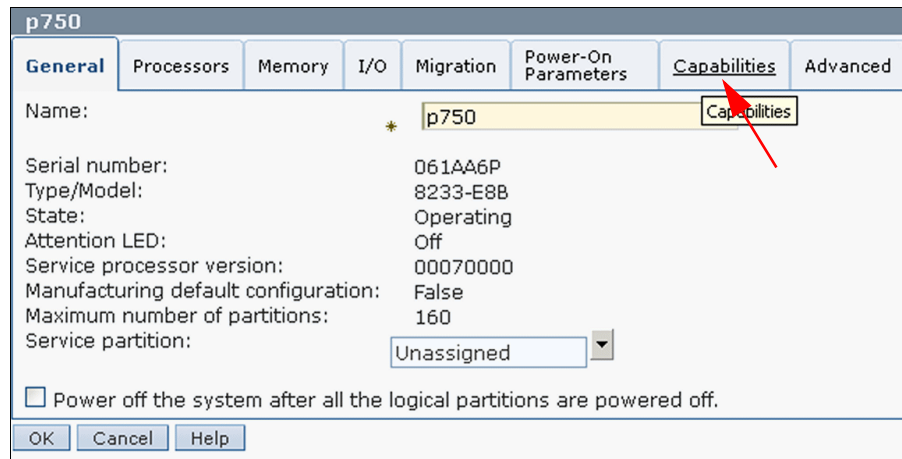
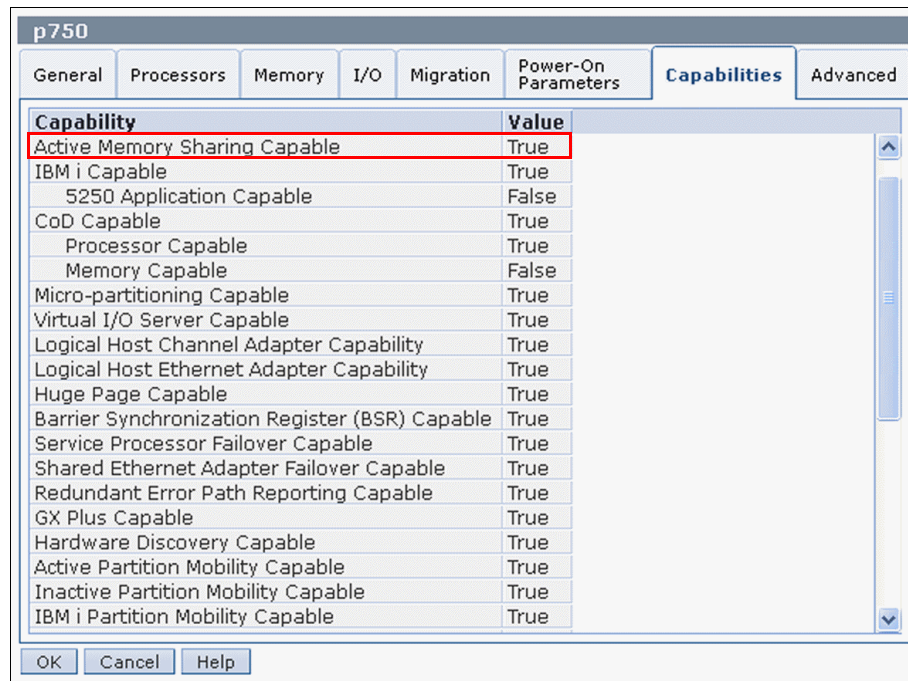


Figure 9-7 Selecting Capabilities tab in HMC managed system Properties

4. In the **Capability** list for the managed system, find the **Active Memory Sharing Capable** field, as shown in Figure 9-8. If its value is True, the managed system is Active Memory Sharing capable, and IBM PowerVM Enterprise edition is enabled on your system.



The screenshot shows the 'p750' system configuration window with the 'Capabilities' tab selected. The 'Active Memory Sharing Capable' field is highlighted with a red box, showing a value of 'True'. Below it is a scrollable list of other capabilities and their values.

Capability	Value
Active Memory Sharing Capable	True
IBM i Capable	True
5250 Application Capable	False
CoD Capable	True
Processor Capable	True
Memory Capable	False
Micro-partitioning Capable	True
Virtual I/O Server Capable	True
Logical Host Channel Adapter Capability	True
Logical Host Ethernet Adapter Capability	True
Huge Page Capable	True
Barrier Synchronization Register (BSR) Capable	True
Service Processor Failover Capable	True
Shared Ethernet Adapter Failover Capable	True
Redundant Error Path Reporting Capable	True
GX Plus Capable	True
Hardware Discovery Capable	True
Active Partition Mobility Capable	True
Inactive Partition Mobility Capable	True
IBM i Partition Mobility Capable	True

Figure 9-8 Managed system capabilities in the HMC

Checking system capabilities using HMC CLI tools

You can check the managed system capabilities using the HMC `lssyscfg` command. The syntax of the command is as follows:

```
lssyscfg -r sys -m <managed system> -F name,capabilities
```

In Example 9-1, the items in bold show that the system is Active Memory Sharing and Active Memory Deduplication capable.

Example 9-1 Displaying managed system capabilities using the HMC CLI

```
hscroot@hmc6:~> lssyscfg -r sys -m POWER7_MemoryDedup -F
name,capabilities
POWER7_MemoryDedup,"active_lpar_mobility_capable,inactive_lpar_mobility
_capable,active_lpar_share_idle_procs_capable,active_mem_dedup_capable,
active_mem_expansion_capable,active_mem_sharing_capable,autorecovery_po
wer_on_capable,bsr_capable,cod_proc_capable,custom_mac_addr_capable,ele
ctronic_err_reporting_capable,firmware_power_saver_capable,hardware_pow
er_saver_capable,hardware_discovery_capable,hca_capable,huge_page_mem_c
apable,lhea_capable,lpar_affinity_group_capable,lpar_avail_priority_cap
able,lpar_proc_compat_mode_capable,lpar_remote_restart_capable,lpar_sus
pend_capable,os400_lpar_suspend_capable,micro_lpar_capable,os400_capabl
e,os400_net_install_capable,redundant_err_path_reporting_capable,shared
_eth_failover_capable,sp_failover_capable,vet_activation_capable,virtua
l_eth_dlpar_capable,virtual_eth_qos_capable,virtual_fc_capable,virtual_
io_server_capable,virtual_switch_capable,vlan_stat_capable,vtpm_capable
"
```

Virtual I/O Server version

From the Virtual I/O Server (VIOS), shell as the *padmin* user, issue the `ioslevel` command to check your Virtual I/O Server version. See Example 9-2.

Example 9-2 Checking the Virtual I/O Server version

```
$ ioslevel
2.2.1.1
```

Operating system

This section describes how to check the prerequisites for each of the three operating systems you are able to run on Power: AIX, IBM i, and Linux.

AIX

To check the version of your AIX operating system, use the **oslevel -s** command, as shown in Example 9-3.

Example 9-3 Checking the operating system version

```
root@aix1dedup / # oslevel -s
7100-01-01-1141
```

IBM i

In IBM i, use the **DSPPTF** command to check the operating system's version, as shown in Figure 9-9.

MAIN	IBM i Main Menu	System:
X001AA6P		
Select one of the following:		
1. User tasks		
2. Office tasks		
3. General system tasks		
4. Files, libraries, and folders		
5. Programming		
6. Communications		
7. Define or change the system		
8. Problem handling		
9. Display a menu		
11. IBM i Access tasks		
90. Sign off		
Selection or command		
==> DSPPTF		
F3=Exit	F4=Prompt	F9=Retrieve
F12=Cancel	F23=Set initial	menu

Figure 9-9 IBM i DSPPTF command

Figure 9-10 illustrates the result of the **DSPPTF** command.

```

                                Display PTF Status
                                System:
X001AA6P
Product ID . . . . . : 5770999
IPL source . . . . . : ##MACH#A
Release of base option . . . . . : V7R1M0 L00

Type options, press Enter.
  5=Display PTF details  6=Print cover letter  8=Display cover
letter

      PTF
Opt ID  Status
RE11067 Permanently applied
RE10187 Permanently applied
RE10084 Permanently applied
RE10026 Permanently applied
QLL2924 Permanently applied
MF99002 Permanently applied
MF99001 Permanently applied
MF52514 Permanently applied
      IPL
      Action
      None
      None
      None
      None
      None
      None
      None
      None

More...
F3=Exit  F11=Display alternate view  F17=Position to  F12=Cancel

```

Figure 9-10 IBM i DSPPTF command result

Linux

To check the version of your Linux operating system, use the **cat /etc/redhat-release** command in RedHat Linux (shown in Example 9-4).

Example 9-4 Displaying the RedHat Linux version

```

[root@linux1 ~]# cat /etc/redhat-release
Red Hat Enterprise Linux Server release 6.2 GA (Santiago)

```

You can use the **cat /etc/SuSE-release** command on SuSE SLES Linux (shown in Example).

Example 9-5 Displaying the SLES version

```
linux11:~ # cat /etc/SuSE-release
SUSE Linux Enterprise Server 11 (ppc64)
VERSION = 11
PATCHLEVEL = 2
```

9.2.2 Sizing your systems for Active Memory Deduplication

The following sections discuss sizing considerations.

Memory sizing requirements

The amount of memory used for page deduplication control is managed directly by the hypervisor and is not taken from the shared memory pool. This fact implies that the memory configuration of your LPARs, whether they are Virtual I/O Server, AIX, Linux, or IBM i clients, does not need to be modified when you turn on Active Memory Deduplication for your shared memory LPARs.

The amount of memory used by the hypervisor for page deduplication control can be computed with the following formula:

$$\text{deduplication table size} = \text{AMS max pool size} \times \text{deduplication table ratio}$$

You can tune both the Active Memory Sharing maximum pool size and the deduplication table ratio to manipulate this amount of memory. This principle is explained in further detail in *IBM PowerVM Virtualization Managing and Monitoring*, SG24-7590. However, the memory savings from deduplicating memory, even in a small environment, more than compensate for this memory consumption.

LPAR memory: The memory configuration of your existing shared memory LPARs does not need to be revisited to implement Active Memory Deduplication in your environment:

Processor sizing requirements

The processing power required to deduplicate memory pages is donated by the Virtual I/O Server when they are idle. Therefore, size them with an appropriate slack (additional capacity) to ensure that they have idle cycles to donate.

As a rule of thumb, start by allowing your Virtual I/O Server to obtain extra processing units from the processor shared pool.

- ▶ Uncap your Virtual I/O Server.
- ▶ Set them to the highest uncapped weight of 255.

Then, to create an initial slack for memory deduplication, make the number of virtual processors greater than the amount of desired processing units by at least 0.1 units. Here are several configuration examples for a Virtual I/O Server:

- ▶ If the number of configured desired processing units is between 0.1 and 0.9, make the number of virtual processors at least 1.
- ▶ If the number of configured desired processing units is between 1.0 and 1.9, make the number of virtual processors at least 2.
- ▶ If the number of configured desired processing units is between 2.0 and 2.9, make the number of virtual processors at least 3.

Keep in mind that this process does not consider any analysis of the workload type, average amount of processor units available in the shared processor pool, or any other factor. This way, you create some initial slack in the Virtual I/O Server so that you can turn on Active Memory Deduplication without compromising your current Virtual I/O Server load. In most cases, having a slack of 0.1 processing units is enough.

As you can see, sizing your existing shared memory environment to enable Active Memory Deduplication requires very small changes in resource requirements. If you already have a 0.1 processing unit slack in your Virtual I/O Server, then you really do not need to do anything but turn on Active Memory Deduplication.



I/O virtualization planning

This chapter gives you the necessary planning details involved in I/O virtualization.

It covers the following topics:

- ▶ Virtual I/O Server planning
- ▶ Storage virtualization planning
- ▶ Network virtualization planning

10.1 Virtual I/O Server planning

This section discusses details to be considered for planning a Virtual I/O Server.

10.1.1 Specifications required to create the Virtual I/O Server

To activate the Virtual I/O Server, the PowerVM Editions (or Advanced POWER Virtualization) hardware feature is required. A logical partition with enough resources to share with other logical partitions is also required. Table 10-1 shows a list of minimum hardware requirements that must be available to create the Virtual I/O Server.

Table 10-1 Resources that are required

Resource	Requirement
Hardware Management Console or Integrated Virtualization Manager	The HMC or Integrated Virtualization Manager is required to create the logical partition and assign resources.
Storage adapter	The server logical partition needs at least one storage adapter.
Physical disk	The disk must be at least 30 GB, This disk can be shared.
Ethernet adapter	If you want to route network traffic from virtual Ethernet adapters to Shared Ethernet Adapter, you need an Ethernet Adapter.
Memory	For POWER7 processor-based systems, at least 768 MB of memory is required.
Processor	At least 0.05 processor use is required.

Sizing of processor and memory

The sizing of the processor and memory resources for Virtual I/O Server depends on the amount and type of workload that the Virtual I/O Server has to process. For example, network traffic going through a Shared Ethernet adapter requires more processor resource than virtual SCSI traffic (Table 10-2).

Rules: The following examples are *only rules of thumb* and can be used as a starting point when setting up an environment using the Virtual I/O Server for the first time.

Table 10-2 Virtual I/O Server sizing examples

Environment	CPU	Memory
Small environment	0.25 - 0.5 processors (uncapped)	2 GB
Large environment	1 -2 processors (uncapped)	4 GB
Environment using shared storage pools	At least one processor (uncapped)	4 GB

Monitoring: When the environment is in production, the processor and memory resources on the Virtual I/O Server have to be monitored regularly and adjusted if necessary to make sure the configuration fits with workload. More information about monitoring CPU and memory on the Virtual I/O Server can be found in the publication, *IBM PowerVM Virtualization Managing and Monitoring*, SG24-7590-01, at this website:

<http://www.redbooks.ibm.com/abstracts/sg247590.html>

For detailed sizing information and guidelines, see the Virtual I/O Server capacity planning section in the IBM Power Systems Hardware Information Center:

http://publib.boulder.ibm.com/infocenter/powersys/v3r1m5/topic/p7hb1/ipb1_vios_planning_cap.htm

The Virtual I/O Server is designed for selected configurations that include specific models of IBM and other vendor storage products.

Consult your IBM representative or Business Partner for the latest information and included configurations.

List of supported adapters and storage devices

Virtual devices exported to client partitions by the Virtual I/O Server must be attached through supported adapters. An updated list of supported adapters and storage devices is available at:

<http://www14.software.ibm.com/webapp/set2/sas/f/vios/documentation/data-sheet.html>

Plan carefully before you begin the configuration and installation of your Virtual I/O Server and client partitions. Depending on the type of workload and needs of an application, it is possible to mix virtual and physical devices in the client partitions.

For further information about the Virtual I/O Server, including planning information, see the IBM Systems Hardware Information Center at this website:

<http://publib.boulder.ibm.com/infocenter/powersys/v3r1m5/index.jsp?topic=/p7hb1/iphb1kickoff.html>

Planning considerations while mapping storage from Virtual I/O Server

As discussed in the overview part, storage can be mapped from Virtual I/O Server in different ways. The following sections provide more details about mapping storage.

Mapping physical volumes

The Virtual I/O Server can export physical volumes intact to virtual I/O clients. This method of exporting storage has several advantages over logical volumes:

- ▶ Physical disk devices can be exported from two or more Virtual I/O Servers concurrently for multipath redundancy.
- ▶ The code path for exporting physical volumes is shorter, which might lead to better performance.
- ▶ Physical disk devices can be moved from one Virtual I/O Server to another with relative ease.
- ▶ In certain cases, existing LUNs from physical servers can be migrated into the virtual environment with the data intact.
- ▶ One consideration for exporting physical volumes is that the size of the device is not managed by the Virtual I/O Server, and the Virtual I/O Server does not allow partitioning of a single device among multiple clients. This is generally only a concern for internal and SCSI-attached disks.

There is no general requirement to subdivide SAN-attached disks because storage allocation can be managed at the storage server. In the SAN environment, provision and allocate LUNs for each LPAR on the storage servers and export them from the Virtual I/O Server as physical volumes.

When a SAN disk is available, all storage associated with a virtual I/O client should be stored in the SAN, including rootvg and paging space. This makes management simpler because partitions will not be dependent on both internal logical volumes and external LUNs. It also makes it easier to move virtual servers from one Virtual I/O Server to another.

Logical volumes

The Virtual I/O Server can export logical volumes to virtual I/O clients. This method does have some advantages over physical volumes:

- ▶ Logical volumes can subdivide physical disk devices between clients.
- ▶ The logical volume interface is familiar to those who have AIX experience.

Important: The rootvg on the Virtual I/O Server should not be used to host exported logical volumes because manual intervention might be required. Certain types of software upgrades and system restores might alter the logical volume to target device mapping for logical volumes within rootvg.

When an internal or SCSI-attached disk is used, the logical volume manager (LVM) enables disk devices to be subdivided between virtual I/O clients. For small servers, this enables several virtual servers to share internal disks or RAID arrays.

A logical volume on the Virtual I/O Server used as a virtual SCSI disk cannot exceed 1 TB in size.

Important: Although logical volumes that span multiple physical volumes are possible, for optimum performance, a logical volume has to reside wholly on a single physical volume. To guarantee this, volume groups can be composed of single physical volumes.

Keeping an exported storage pool backing device or logical volume on a single hdisk results in optimized performance.

Bad Block Relocation on the Virtual I/O Server Version 2.1.2 and above is supported:

- ▶ As long as a virtual SCSI device is not striped
- ▶ As long as a virtual SCSI device is not mirrored
- ▶ When the logical volume wholly resides on a single physical volume

Although supported, Bad Block Relocation must not be enabled on the Virtual I/O Server for virtual SCSI devices. Bad Block Relocation has to be enabled for virtual SCSI devices on the clients to obtain better performance, such as for a virtual tape device. Bad Block Relocation needs to be used for paging spaces used by Active Memory Sharing (AMS). Use the `chlv` command to change the Bad Block Relocation policy for logical volumes on the Virtual I/O Server.

To verify that a logical volume does not span multiple disks, run the `lslv` command as shown here:

```
$ lslv -pv app_vg
app_vg:N/A
PV          COPIES      IN BAND      DISTRIBUTION
hdisk5      320:000:000    99%         000:319:001:000:000
```

Only one disk must appear in the resulting list.

Best practices for exporting logical volumes

The Integrated Virtualization Manager (IVM) and HMC-managed environments present two separate interfaces for storage management under different names. The storage pool interface under the IVM is essentially the same as the logical volume manager interface under the HMC, and in some cases, the documentation uses the terms interchangeably. The remainder of this chapter uses the term *volume group* to refer to both volume groups and storage pools, and the term *logical volume* to refer to both logical volumes and storage pool backing devices.

Tip: The storage pool commands are also available on the HMC.

Logical volumes enable the Virtual I/O Server to subdivide a physical volume between multiple virtual I/O clients. In many cases, the physical volumes used will be internal disks, or RAID arrays built of internal disks.

A single volume group should not contain logical volumes used by virtual I/O clients and logical volumes used by the Virtual I/O Server operating system. Keep Virtual I/O Server file systems within the rootvg, and use other volume groups to host logical volumes for virtual I/O clients.

A single volume group or logical volume cannot be accessed by two Virtual I/O Servers concurrently. Do not attempt to configure MPIO on virtual I/O clients for VSCSI devices that reside on logical volumes. If redundancy is required in logical volume configurations, use LVM mirroring on the virtual I/O client to mirror across logical volumes on different Virtual I/O Servers.

Although logical volumes that span multiple physical volumes are supported, a logical volume should reside wholly on a single physical volume for optimum performance. To guarantee this, volume groups can be composed of single physical volumes.

Remember: Keeping an exported storage pool backing device or logical volume on a single hdisk results in optimized performance.

When exporting logical volumes to clients, the mapping of individual logical volumes to virtual I/O clients is maintained in the Virtual I/O Server. The additional level of abstraction provided by the logical volume manager makes it important to track the relationship between physical disk devices and virtual I/O clients.

Storage pools

When managed by the Integrated Virtualization Manager (IVM), the Virtual I/O Server can export storage pool backing devices to virtual I/O clients. This method is similar to logical volumes, and has some advantages over physical volumes:

- ▶ Storage pool backing devices can subdivide physical disk devices between separate clients.
- ▶ The storage pool interface is easy to use through IVM.

Important: The default storage pool in IVM is the root volume group of the Virtual I/O Server. Be careful not to allocate backing devices within the root volume group because certain types of software upgrades and system restores might alter the logical volume to target device mapping for logical volumes in rootvg, requiring manual intervention.

Systems in a single server environment under the management of IVM are often not attached to a SAN, and these systems typically use internal and SCSI-attached disk storage. The IVM interface allows storage pools to be created on physical storage devices so that a single physical disk device can be divided among several virtual I/O clients.

As with logical volumes, storage pool backing devices cannot be accessed by multiple Virtual I/O Servers concurrently, so they cannot be used with MPIO on the virtual I/O client.

If redundancy is required, use LVM mirroring on the virtual I/O client.

File-backed devices

Starting with Version 1.5 of Virtual I/O Server, there is a feature called *file-backed virtual SCSI devices*. This feature provides additional flexibility for provisioning and managing virtual SCSI devices. In addition to backing a virtual SCSI device (disk or optical) by physical storage, a virtual SCSI device can be backed by a file. File-backed virtual SCSI devices continue to be accessed as standard SCSI-compliant storage.

Remember: If LVM mirroring is used in the client, make sure that each mirror copy is placed on a separate disk, and mirrored across storage pools.

Logical units in shared storage pool

Starting with Virtual I/O Server Version 2.2.0.11, Fix Pack 24 Service Pack 1, or later, there is a new feature called *logical units*. This feature provides flexibility for provisioning and better storage utilization for disk devices being accessed through a SAN.

Logical units are created in a shared storage pool, can be thin or thick provisioned and are accessed as standard SCSI storage on a client partition. Virtual disks from a shared storage pool support the persistent reservation.

The shared storage pool has to be defined on a Virtual I/O Server cluster specific to this purpose.

Starting with version 2.2.2.0, a Virtual I/O Server cluster can have 16 nodes, and storage devices can be shared among them to create a shared storage pool with a maximum capacity of 512 TB. Supported sizes for the logical units allocated from the pool are from 1 GB to 4 TB. In case of thin provisioned disks only a minimal amount will be used on the physical disks however if the logical unit is created as thick provisioned it will instantly occupy the space on the disk according to its defined size.

10.1.2 Redundancy considerations

This section discusses requirements for providing high availability for Virtual I/O Servers.

Several considerations are taken into account when deciding to use multiple Virtual I/O Servers. Client uptime, predicted I/O load averages for both network and storage, and system manageability are areas requiring consideration when planning the Virtual I/O Server configuration. With PowerVM Standard or PowerVM Enterprise edition, multiple Virtual I/O Servers provide the performance and redundancy expected for those types of environment.

For a small system with limited resources and limited I/O adapters, the system might not have enough resources for a second Virtual I/O Server. With limited I/O adapters, having a second Virtual I/O Server might adversely affect the overall performance provided to the clients if the additional adapters are used for increasing redundancy, not throughput.

For a larger system, there is a lower resource constraint, and multiple Virtual I/O Servers can be deployed without affecting overall client performance. More I/O adapters cater for both throughput and redundancy when used with additional Virtual I/O Servers.

The Virtual I/O Server is extremely robust. It mainly runs device drivers and does not run any application workloads, and regular users do not log in. Redundancy can be built into the Virtual I/O Server itself by using redundant physical adapters combined with MPIO or LVM mirroring for storage devices and Link Aggregation for network devices.

In a dual Virtual I/O Server configuration, virtual SCSI and Shared Ethernet Adapter can be configured in a redundant fashion allowing system maintenance such as reboot, software updates or even reinstallation to be performed on a Virtual I/O Server without affecting the virtual I/O clients. This is the main reason to implement two Virtual I/O Servers.

With the proper planning and architecture implementation, maintenance can now be performed, not only on a Virtual I/O Server, but also on any external device it connects to, such as a network or SAN switch, removing the layer of physical resource dependency.

With the client partition using multipathing and using SEA failover, no actions will need to be performed on the client partition while the system maintenance is being performed or after it has completed. This results in improved uptime and reduced system administration efforts for the client partitions.

Tip: A combination of multipathing for disk redundancy and Shared Ethernet Adapter failover for network redundancy is best.

Upgrading and rebooting a Virtual I/O Server, network switch, or SAN switch is simpler and more compartmentalized, because the client is no longer dependent on the availability of all of the environment.

Support: IVM supports a single Virtual I/O Server.

In Figure 10-1, a client partition has virtual SCSI devices and a virtual Ethernet adapter hosted from two Virtual I/O Servers. The client has multipathing implemented across the virtual SCSI devices and Shared Ethernet Adapter failover for the virtual Ethernet.

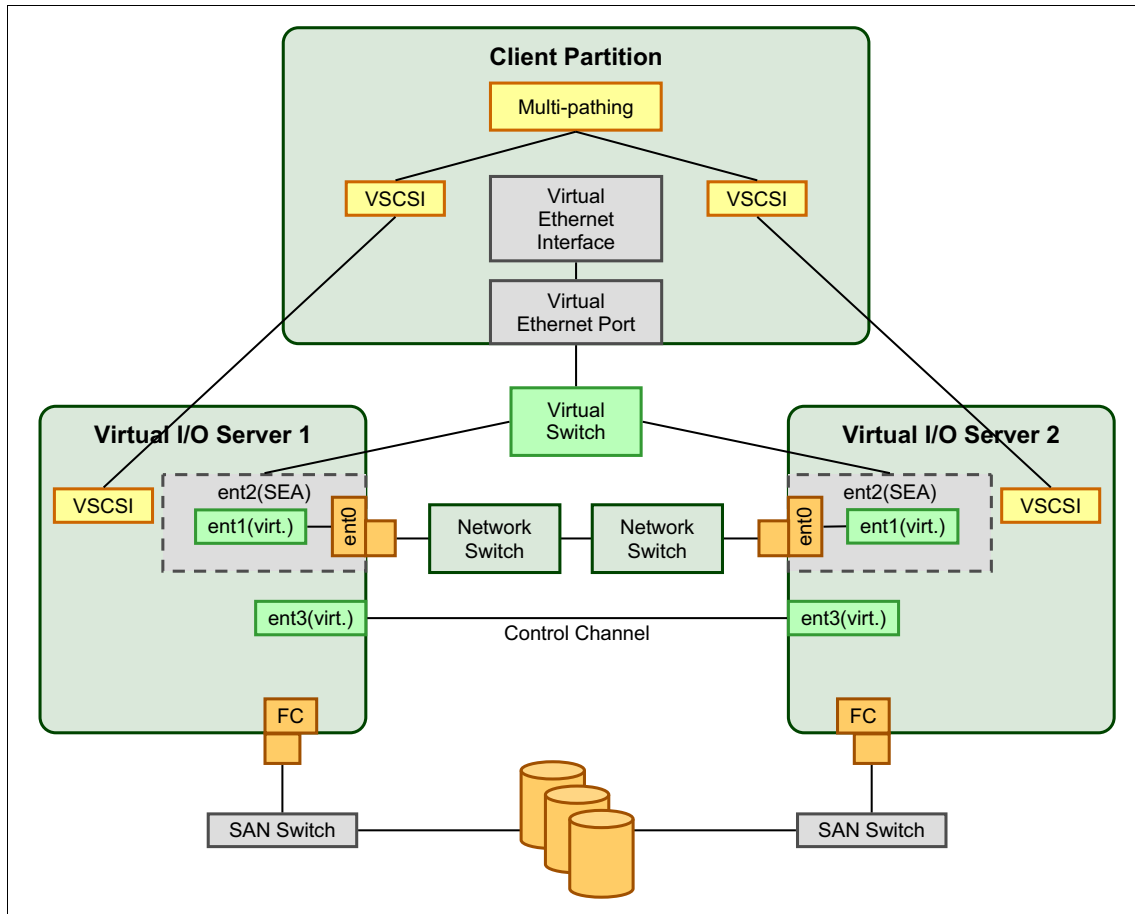


Figure 10-1 Redundant Virtual I/O Servers before maintenance

When Virtual I/O Server 2 is shut down for maintenance, as shown in Figure 10-2, the client partition continues to access the network and SAN storage through Virtual I/O Server 1.

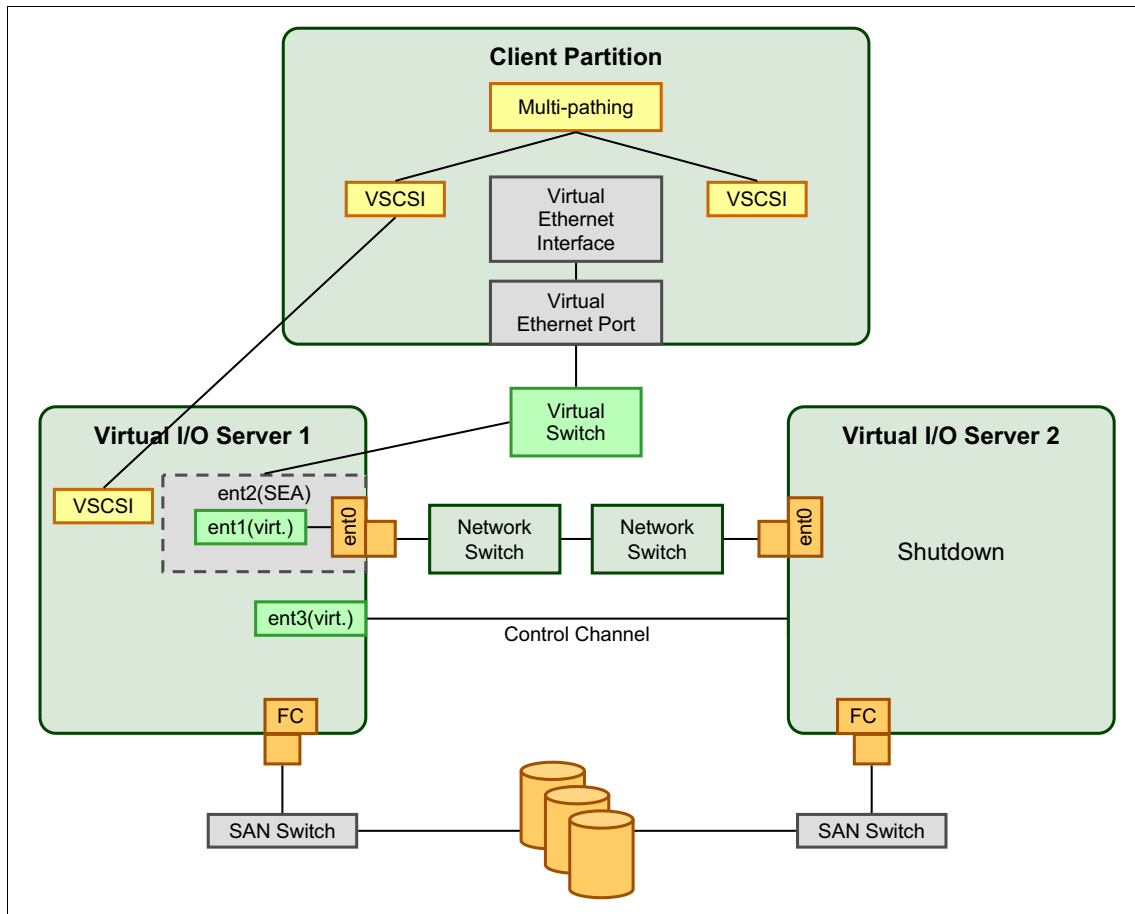


Figure 10-2 Redundant Virtual I/O Servers during maintenance

When Virtual I/O Server 2 returns to a full running state, these events occur:

- ▶ An AIX client will continue using the MPIO path through Virtual I/O Server 1, unless the MPIO path is manually changed to Virtual I/O Server 2.
- ▶ An IBM i or Linux multipathing client, which uses a round-robin multipathing algorithm, will automatically start using both paths when the path to Virtual I/O Server 2 becomes operational again.
- ▶ If Virtual I/O Server 2 is the primary Shared Ethernet Adapter, client network traffic going thru the backup Shared Ethernet Adapter on Virtual I/O Server 1 will automatically resume on Virtual I/O Server 2.

In addition to continuous availability, a dual Virtual I/O Server setup also provides the ability to separate or balance the virtual I/O load and the resulting resource consumption across the Virtual I/O Servers.

Virtual Ethernet traffic is generally heavier on the Virtual I/O Server than virtual SCSI traffic. Virtual Ethernet connections generally take up more CPU cycles than connections through physical Ethernet adapters. The reason is that modern physical Ethernet adapters contain many functions to off-load some work from the system's CPUs, for example, checksum computation and verification, interrupt modulation, and packet reassembly.

In a configuration running MPIO and a single SEA per Virtual I/O Server you typically separate the traffic in such a way that the virtual Ethernet traffic is going through one Virtual I/O Server and the virtual SCSI traffic is going through the other. This is done by defining the SEA trunk priority and the MPIO path priority. See 16.3.2, "SEA failover" on page 591 for details on SEA failover configuration and 16.2.5, "Availability" on page 493 for details on MPIO configuration.

Important: Do not switch off SEA threading on the Virtual I/O Server that is used as primary for the network traffic. It will do both disk and network serving in case the other Virtual I/O Server is unavailable.

Tip: By default, MPIO traffic is normally routed through the first Virtual I/O Server. The easiest way to separate disk and network traffic is therefore to define the Shared Ethernet Adapter to use the second Virtual I/O Server as primary channel. The path priorities for the virtual hdisk can be left unchanged. Otherwise you have to update the path priorities for each hdisk in each client partition.

In an MPIO configuration with several Shared Ethernet Adapters per Virtual I/O Server, you typically balance the network and virtual SCSI traffic between the Virtual I/O Servers.

Paths: Use the storage configuration software to check that the preferred paths on the storage subsystem are in accordance with the path priorities set in the virtual I/O clients.

When using MPIO together with NIB, the virtual Ethernet traffic can be spread more flexibly. For each network interface running in NIB mode, you can configure through which Virtual I/O Server the traffic must go. If your network traffic is driving one Virtual I/O Server to its limits, you can balance the workloads by adapting the NIB configuration.

If the Virtual I/O Servers are running in uncapped mode, processor resources are automatically balanced.

See 16.3.3, “EtherChannel Backup in the AIX client” on page 603 for details on configuring NIB and 16.2.5, “Availability” on page 493 for details on MPIO configuration.

Figure 10-3 shows an example configuration where network and disk traffic are separated:

- ▶ Virtual I/O Server 1 has priority 1 for network and priority 2 for disk.
- ▶ Virtual I/O Server 2 has priority 2 for network and priority 1 for disk.

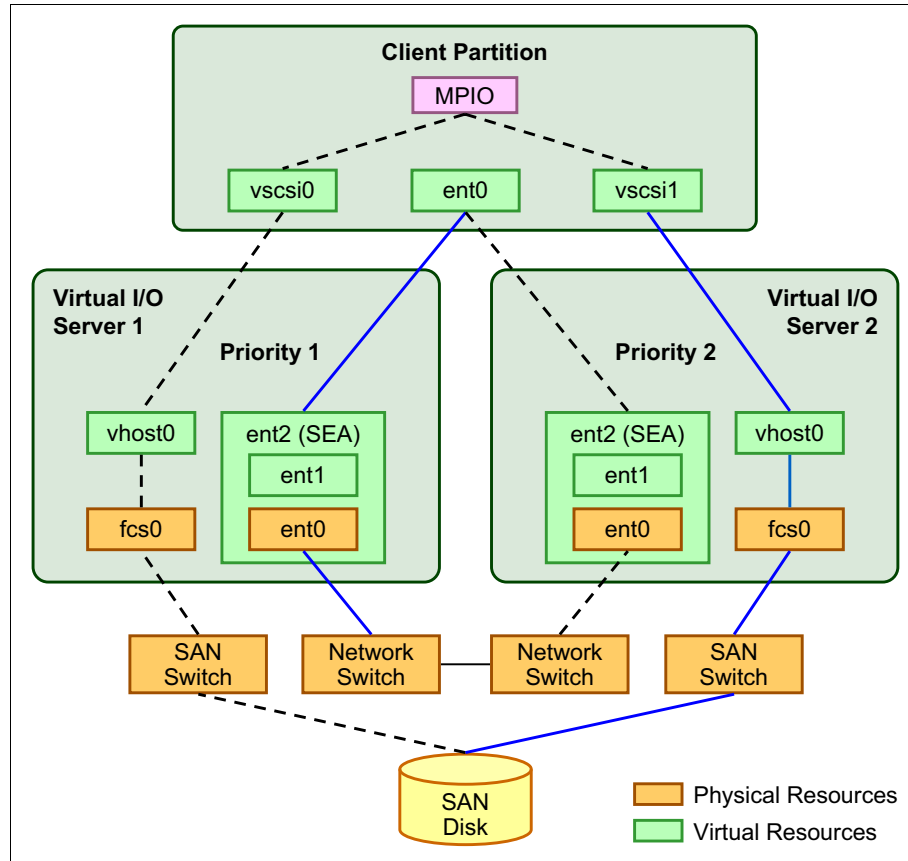


Figure 10-3 Separating disk and network traffic

10.2 Storage virtualization planning

The following sections explain how to plan storage virtualization in a PowerVM environment.

10.2.1 Virtual SCSI

This section talks about planning Virtual SCSI in PowerVM:

- ▶ Virtual SCSI supports Fibre Channel, parallel SCSI, SCSI RAID devices, and optical devices, including DVD-RAM and DVD-ROM. Other protocols, such as SSA and tape devices, are not supported.
- ▶ The SCSI protocol defines mandatory and optional commands. While virtual SCSI supports all the mandatory commands, not all optional commands are supported.

Migration to Virtual SCSI environment

A storage device can be moved from physically connected SCSI to a virtually connected SCSI device if it meets the following criteria:

- ▶ The device is an entire physical volume (for example, a LUN).
- ▶ The device capacity is identical in both physical and virtual environments.
- ▶ The Virtual I/O Server is able to manage the device using a UDID or iEEE ID.

Support: Physical to virtual (p2v) migration is not supported for IBM i storage devices due to the unique 520 byte sector size that IBM i uses natively.

Devices managed by the following multipathing solutions within the Virtual I/O Server are expected to be UDID devices:

- ▶ All multipath I/O (MPIO) versions, including Subsystem Device Driver Path Control Module (SDDPCM), EMC PCM, and Hitachi Dynamic Link Manager (HDLM) PCM
- ▶ EMC PowerPath 4.4.2.2 or later
- ▶ IBM Subsystem Device Driver (SDD) 1.6.2.3 or later
- ▶ Hitachi HDLM 5.6.1 or later

Virtual SCSI devices created with earlier versions of PowerPath, HDLM, and SDD are not managed by UDID format and are not expected to be p2v compliant. The operations mentioned before (for example, data replication or movement between Virtual I/O Server and non-Virtual I/O Server environments) are not likely to work in these cases.

The **chkdev** command can be used to verify if a device that is attached to a Virtual I/O Server can be migrated from a physical adapter to using virtual adapter. Example 10-1 shows a LUN provided by an external storage subsystem. As you can see, it shows YES in the PHYS2VIRT_CAPABLE field. Therefore it can be migrated to a virtual device simply by mapping it to a virtual SCSI server adapter.

Example 10-1 Using chkdev to verify p2v compliance

```
$ chkdev -dev hdisk4 -verbose
NAME:                hdisk4
IDENTIFIER:          200B75BALB1101207210790003IBMfcp
PHYS2VIRT_CAPABLE:  YES
VIRT2NPIV_CAPABLE:   NA
VIRT2PHYS_CAPABLE:   NA
PVID:                00f61aa66682ec6800000000000000000
UDID:                200B75BALB1101207210790003IBMfcp
IEEE:
VTD:
```

For more information, see the IBM Power Systems Hardware Information Center at this website:

http://publib.boulder.ibm.com/infocenter/powersys/v3r1m5/topic/p7hb1/ip_hb1_vios_device_compat.htm

For a description of physical to virtual storage migration, see the publication, *PowerVM Migration from Physical to Virtual Storage*, SG24-7825-00, available at the following website:

<http://www.redbooks.ibm.com/abstracts/sg247825.html>

Performance considerations

Provided that there is sufficient CPU processing capacity available, the performance of virtual SCSI has to be comparable to dedicated I/O devices.

Virtual Ethernet, having non-persistent traffic, runs at a higher priority than the virtual SCSI on the VIO server. To make sure that high volumes of networking traffic will not starve virtual SCSI of CPU cycles, a threaded mode of operation has been implemented for the Virtual I/O Server by default since version 1.2.

Maximum number of slots

Virtual SCSI itself does not have any maximums in terms of number of supported devices or adapters. The Virtual I/O Server supports a maximum of 1024 virtual I/O slots per Virtual I/O Server. A maximum of 256 virtual I/O slots can be assigned to a single client partition.

Every I/O slot needs some physical server resources to be created. Therefore, the resources assigned to the Virtual I/O Server puts a limit on the number of virtual adapters that can be configured.

Installation and migration considerations

These are the major installation and migration considerations:

- ▶ Consider the client partition root volume group sizes prior to creating logical volumes when using AIX levels lower than AIX 6.1 Technology Level 4.
- ▶ Increasing a rootvg by extending its associated Virtual I/O Server logical volume is only supported on AIX 6.1 Technology Level 4 or later.

Naming conventions

A well-planned naming convention is key in managing the information. One strategy for reducing the amount of data that must be tracked is to make settings match on the virtual I/O client and server wherever possible.

This can include corresponding volume group, logical volume, and virtual target device names. Integrating the virtual I/O client host name into the virtual target device name can simplify tracking on the server.

When using Fibre Channel disks on a storage server that supports LUN naming, this feature can be used to make it easier to identify LUNs. Commands such as **pcmpath query device** for the IBM System Storage DS8000 and IBM SAN Volume Controller or IBM Storwize® V7000 series storage servers, and the **fget_config** or **mpio_get_config** command for the IBM DS4000 series, can be used to match hdisk devices with LUN IDs or names.

In many cases, using LUN names can be simpler than tracing devices using Fibre Channel world wide port names and numeric LUN identifiers.

The Virtual I/O Server version 2.1 uses MPIO as a default device driver. Example 10-2 shows the listing of a DS4800 disk subsystem. Proper user label naming in the SAN makes it much easier to track the LUN-to-hdisk relation.

Example 10-2 SAN storage listing on the Virtual I/O Server version 2.1

```
$ oem_setup_env
# mpio_get_config -Av
Frame id 0:
  Storage Subsystem worldwide name: 60ab800114632000048ed17e
  Controller count: 2
  Partition count: 1
  Partition 0:
    Storage Subsystem Name = 'ITS0_DS4800'
      hdisk      LUN #   Ownership      User Label
```


hdisk6	0	A (preferred)	VIOS1
hdisk7	1	A (preferred)	AIX61
hdisk8	2	B (preferred)	AIX53
hdisk9	3	A (preferred)	SLES10
hdisk10	4	B (preferred)	RHEL52
hdisk11	5	A (preferred)	IBMi61_0
hdisk12	6	B (preferred)	IBMi61_1
hdisk13	7	A (preferred)	IBMi61_0m
hdisk14	8	B (preferred)	IBMi61_1m

Virtual device slot numbers

After you establish the naming conventions, also establish slot numbering conventions for the virtual I/O adapters.

All Virtual SCSI and Virtual Ethernet devices have slot numbers. In complex systems, there will tend to be far more storage devices than network devices because each virtual SCSI device can only communicate with one server or client.

One common example of slot number assignment is to reserve lower slot numbers for Ethernet adapters. To avoid mixing slot number ranges, allow for growth both for Ethernet and SCSI devices. Virtual I/O Servers typically have many more virtual adapters than client partitions.

Remember: Several disks can be mapped to the same server-client SCSI adapter pair.

Management can be simplified by keeping slot numbers consistent between the virtual I/O client and server. However, when partitions are moved from one server to another, this might not be possible. In environments with only one Virtual I/O Server, add storage adapters incrementally starting with slot 21 and higher.

When clients are attached to two Virtual I/O Servers, the adapter slot numbers should be alternated from one Virtual I/O Server to the other. The first Virtual I/O Server should use odd numbered slots starting at 21, and the second should use even numbered slots starting at 22. In a two-server scenario, allocate slots in pairs, with each client using two adjacent slots such as 21 and 22, or 33 and 34.

Set the maximum virtual adapters number to at least 100. As shown in Figure 10-4, the default value is 10 when you create an LPAR. The appropriate number for your environment depends on the number of virtual servers and adapters expected on each system. Each unused virtual adapter slot consumes a small amount of memory, so the allocation should be balanced.

Important: When you plan for the number of virtual I/O slots on your LPAR, the maximum number of virtual adapter slots available on a partition is set by the partition's profile. To increase the maximum number of virtual adapters you must change the profile, stop the partition (not just a reboot), and start the partition.

To add new virtual I/O clients without shutting down the LPAR or Virtual I/O Server partition, leave plenty of room for expansion when setting the maximum number of slots.

The maximum number of virtual adapters should not be set higher than 1024 as it can cause performance problems.

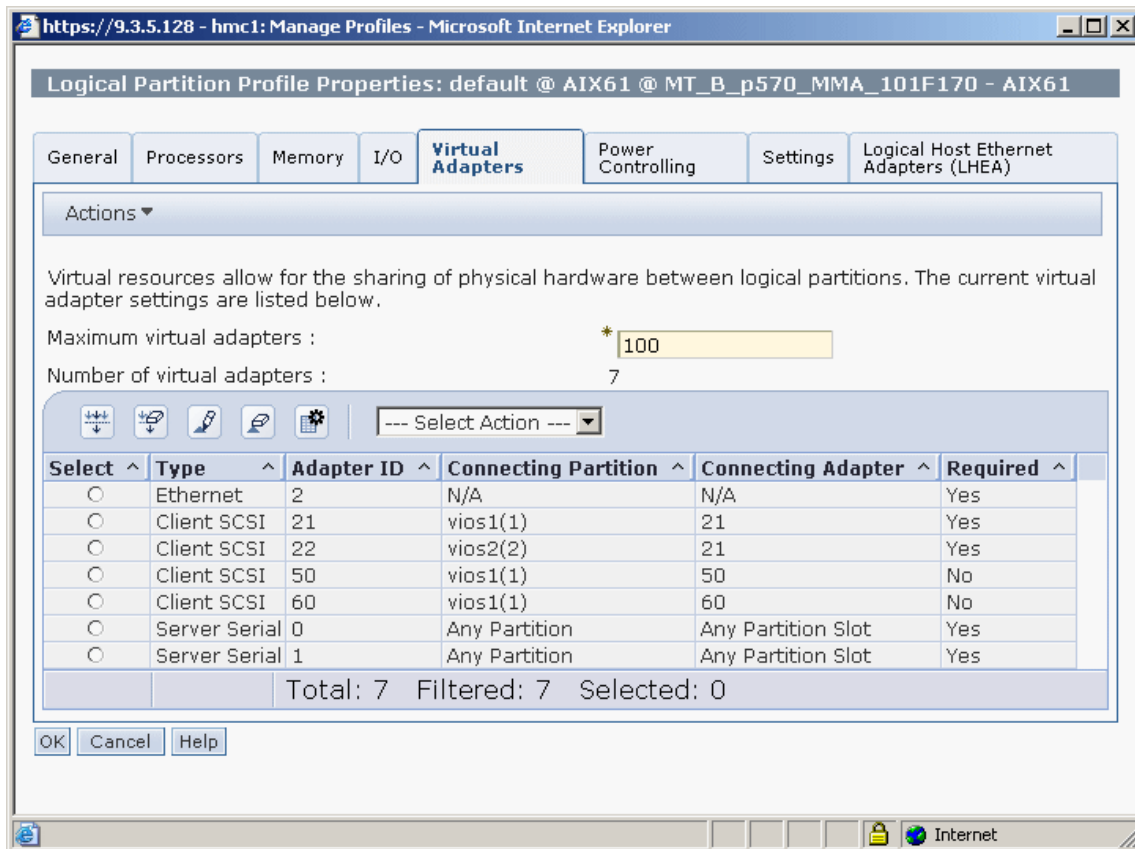


Figure 10-4 Setting maximum number of virtual adapters in a partition profile

Because virtual SCSI connections operate at memory speed, there is generally no performance gain from adding multiple adapters between a Virtual I/O Server and client. However, multiple virtual adapters should be configured when you are using mirroring on the virtual I/O client across multiple storage subsystems for availability, as shown in “AIX client LVM mirroring” on page 540.

- ▶ For AIX virtual I/O client partitions, each adapter pair can handle up to 85 virtual devices with the default queue depth of three.
- ▶ For IBM i clients, up to 16 virtual disk and 16 optical devices are supported.
- ▶ For Linux clients, by default, up to 192 virtual SCSI targets are supported.

In situations where virtual devices per partition are expected to exceed that number, or where the queue depth on certain devices might be increased above the default, reserve additional adapter slots for the Virtual I/O Server and the virtual I/O client partition. When tuning queue depths, the VSCSI adapters have a fixed queue depth. There are 512 command elements, of which 2 are used by the adapter, 3 are reserved for each VSCSI LUN for error recovery, and the rest are used for I/O requests. Thus, with the default queue depth of 3 for VSCSI LUNs, that allows for up to 85 LUNs to use an adapter: $(512 - 2) / (3 + 3) = 85$ rounding down. If you need higher queue depths for the devices, the number of LUNs per adapter is reduced. For example, if you want to use a queue depth of 25, that allows $510/28 = 18$ LUNs per adapter for an AIX client partition.

For Linux clients, the maximum number of LUNs per virtual SCSI adapter is decided by the *max_id* and *max_channel* parameters. The *max_id* is set to 3 by default and can be increased to 7. The *max_channel* is set to 64 by default, which is the maximum value. With the default values, the Linux client can have $3 * 64 = 192$ virtual SCSI targets. Note that if you overload an adapter, your performance will be reduced.

VSCSI storage planning with migration in mind

Managing storage resources during an LPAR move can be more complex than managing network resources. Careful planning is required to ensure that the storage resources belonging to an LPAR are in place on the target system. This section assumes that you have fairly good knowledge of PowerVM Live Partition Mobility. For more details about that subject, see “Live Partition Mobility planning” on page 260.

Virtual adapter slot numbers

Virtual SCSI and virtual Fibre Channel adapters are tracked by slot number and partition ID on both the Virtual I/O Server and client. The number of virtual adapters in a Virtual I/O Server must equal the sum of the virtual adapters in a client partition that it serves. The Virtual I/O Server *vhost* or *vfhost* adapter slot numbers are not required to match the client partition *vscsi* or *fscsi* slot numbers as shown for virtual SCSI in Figure 10-5.

New system plan - Edit Virtual Slots - Microsoft Internet Explorer

IBM System Planning Tool

System: 570 (IBM Power 9117-MMA)

Edit Virtual Slots
Work with virtual adapter slots and mappings

vios (Virtual I/O Server)
Total slots: 100 Used slots: 15 Available slots: 85

Console

Slot	Type	Target Partition	Target Slot
0	Server		
1	Server		

Reserved
Slot 2 through 10 are reserved for system use.

Ethernet

Slot	PVID	Additional VLANs
11	1	

SCSI

Slot	Type	Target Partition	Target Slot
21	Server	aix61 (AIX 6.1)	31
22	Server	IBMi (IBM i V6R1M0)	31
23	Server	aix53 (AIX 5.3)	31

aix61 (AIX 6.1)
Total slots: 100 Used slots: 4 Available slots: 96

Console

Slot	Type	Target Partition	Target Slot
0	Server		
1	Server		

Ethernet

Slot	PVID	Additional VLANs
2	1	

SCSI

Slot	Type	Target Partition	Target Slot
31	Client	vios (Virtual I/O Server)	21

IBMi (IBM i V6R1M0)
Total slots: 100 Used slots: 4 Available slots: 96

Console

Slot	Type	Target Partition	Target Slot
0	Server		
1	Server		

Ethernet

Slot	PVID	Additional VLANs
2	1	

OK Apply Cancel Help

Figure 10-5 Slot numbers that are identical in the source and target system

You can apply any numbering scheme as long as server-client adapter pairs match. To avoid interchanging types and slot numbers, reserve a range of slot numbers for each type of virtual adapter. This is also important when partitions are moved between systems.

Important: Do not increase the maximum number of adapters for a partition beyond 1024.

10.2.2 Virtual Fibre Channel

Based on the concepts presented on 4.2.2, “Virtual Fibre Channel” on page 50 we will discuss the usage of virtual Fibre Channel on the Virtual I/O Servers to allow the Live Partition Mobility.

Role Of Virtual I/O Server

For Virtual Fibre Channel, the Virtual I/O Server acts as an FC pass-through instead of a SCSI emulator such as when using virtual SCSI. Following Figure 10-6.shows the same.

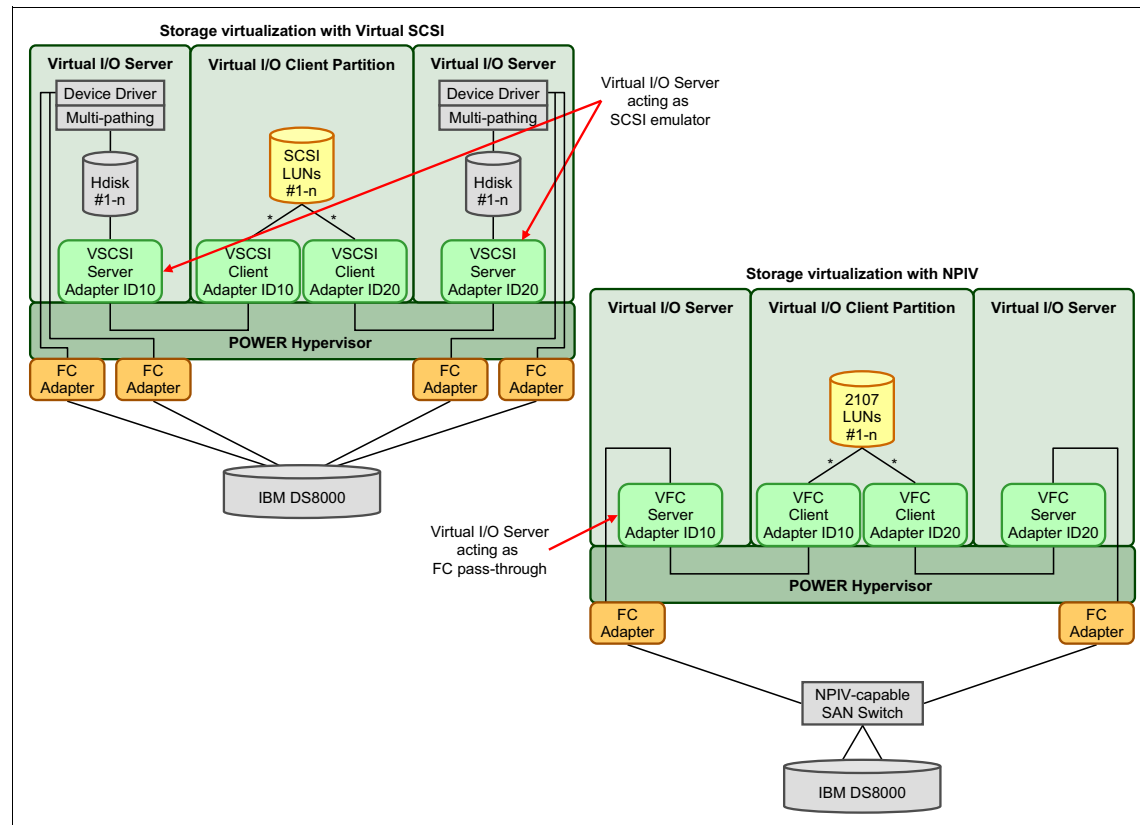


Figure 10-6 Comparing virtual SCSI and Virtual Fibre Channel

Two unique virtual world-wide port names (WWPNs) starting with the letter **c** are generated by the HMC for the VFC client adapter, which after activation of the client partition, log into the SAN like any other WWPNs from a physical port so that disk or tape storage target devices can be assigned to them as if they were physical FC ports.

Tip: Unless using PowerVM Live Partition Mobility or Suspend/Resume, only the first of the two created virtual WWPNs of a VFC client adapter is used.

Requirements

You must meet the following requirements to set up and use Virtual Fibre Channel:

1. Hardware:

- Any POWER6-based system or later.

For Virtual Fibre Channel support on IBM POWER Blades, see the *IBM BladeCenter Interoperability Guide* at this website:

<http://www-947.ibm.com/support/entry/portal/docdisplay?brand=5000020&indocid=MIGR-5073016>

Install a minimum System Firmware level of EL340_039 for the IBM Power 520 and Power 550, and EM340_036 for the IBM Power 560 and IBM Power 570 models.

- Minimum of one 8 Gigabit PCI Express Dual Port Fibre Channel Adapter (feature code 5735, low-profile feature code 5273) or one 10 Gb FCoE PCI Express Dual Port Adapter (feature code 5708, low-profile feature code 5270).

Support: Only the 8 Gigabit PCI Express Dual Port Fibre Channel Adapter (feature code 5735) and the 10 Gb FCoE PCI Express Dual Port Adapter (feature code 5708) are supported for Virtual Fibre Channel.

Install the latest available firmware for the Fibre Channel adapter available at the following IBM Fix Central support website:

<http://www.ibm.com/support/fixcentral>

Important: Establishing a process for regular adapter firmware maintenance is especially important for IBM i customers because the automatic adapter FW update process by IBM i System Licensed Internal Code (SLIC) updates does not apply to any I/O adapters owned by the Virtual I/O Server.

For detailed instructions on how to update the Virtual I/O Server adapter firmware, see *IBM PowerVM Virtualization Managing and Monitoring*, SG24-7590, available at this website:

<http://www.redbooks.ibm.com/abstracts/sg247590.html?Open>

- Virtual Fibre Channel enabled SAN switch:

Only the first SAN switch that is attached to the Fibre Channel adapter in the Virtual I/O Server needs to be Virtual Fibre Channel-capable. Other switches in your SAN environment do not need to be Virtual Fibre Channel-capable.

Conditions: At the time of writing, the following conditions apply:

- ▶ Check with the storage vendor as to whether your SAN switch is Virtual Fibre Channel-enabled.
- ▶ For information about IBM SAN switches, see *Implementing an IBM/Brocade SAN with 8 Gbps Directors and Switches*, SG24-6116, and search for Virtual Fibre Channel.
- ▶ Use the latest supported firmware level for your SAN switch.

2. Software:

- HMC V7.3.4, or later
- Virtual I/O Server Version 2.1 with Fix Pack 20.1, or later
- AIX 5.3 TL9, or later
- AIX 6.1 TL2, or later
- SDD 1.7.2.0 + PTF 1.7.2.2
- SDDPCM 2.2.0.0 + PTF v2.2.0.6
- SDDPCM 2.4.0.0 + PTF v2.4.0.1
- IBM i 6.1.1, or later:
 - Requires HMC V7.3.5, or later, and POWER6 firmware Ex350, or later
 - Support for the 10 Gb FCoE PCI Express Dual Port Adapter (feature codes 5708 and 5270) requires Virtual I/O Server Version 2.2 (Fix Pack 24), or later

- Supports IBM System Storage DS8000 series and selected IBM System Storage tape libraries

See the following IBM i KBS document #550098932 for further requirements and supported storage devices:

<http://www-01.ibm.com/support/docview.wss?uid=nas13b3ed3c69d4b7f25862576b700710198>

- SUSE Linux Enterprise Server 10 SP 3 or later
- Red Hat Enterprise Linux Version 5.4 or later

Planning considerations for Virtual Fibre Channel

The following considerations apply when using Virtual Fibre Channel:

- ▶ One VFC client adapter per physical port per client partition:
Intended to avoid a single point of failure
- ▶ Maximum of 64 active VFC client adapter per physical port:
Might be less due to other Virtual I/O Server resource constraints
- ▶ Maximum of 64 targets per virtual Fibre Channel adapter
- ▶ 32,000 unique WWPN pairs per system platform:
 - Removing adapter does not reclaim WWPNs:
 - Can be manually reclaimed through CLI (`mksyscfg`, `chhwres...`)
 - Or use “`virtual_fc_adapters`” attribute
 - If these resources are exhausted, contact your IBM sales representative or Business Partner representative to purchase an activation code for more.

Implementation considerations

A virtual Fibre Channel server adapter needs to be created by the HMC or IVM for the Virtual I/O Server partition profile that connects to a virtual Fibre Channel client adapter created in the client partition.

Fibre Channel: A virtual Fibre Channel client adapter is a virtual device that provides virtual I/O client partitions with a Fibre Channel connection to a storage area network through the Virtual I/O Server partition.

The Virtual I/O Server partition provides the connection between the virtual Fibre Channel server adapters and the physical Fibre Channel adapters assigned to the Virtual I/O Server partition on the managed system.

10.2.3 Redundancy configurations for virtual Fibre Channel adapters

To implement highly reliable virtual I/O storage configurations, plan the following redundancy configurations to protect your virtual I/O production environment from physical adapter failures as well as from Virtual I/O Server failures.

Host bus adapter redundancy

Similar to virtual SCSI redundancy, virtual Fibre Channel redundancy can be achieved using multipathing or mirroring at the client logical partition. The difference between redundancy with virtual SCSI adapters and the Virtual Fibre Channel technology using virtual Fibre Channel client adapters is that the redundancy occurs at the client, because only the virtual I/O client logical partition recognizes the disk. The Virtual I/O Server is essentially just a Fibre Channel pass-through managing the data transfer through the POWER Hypervisor.

A host bus adapter is a physical Fibre Channel adapter that can be assigned to a logical partition. A host bus adapter (HBA) failover provides a basic level of redundancy for the client logical partition, as shown in Figure 10-7.

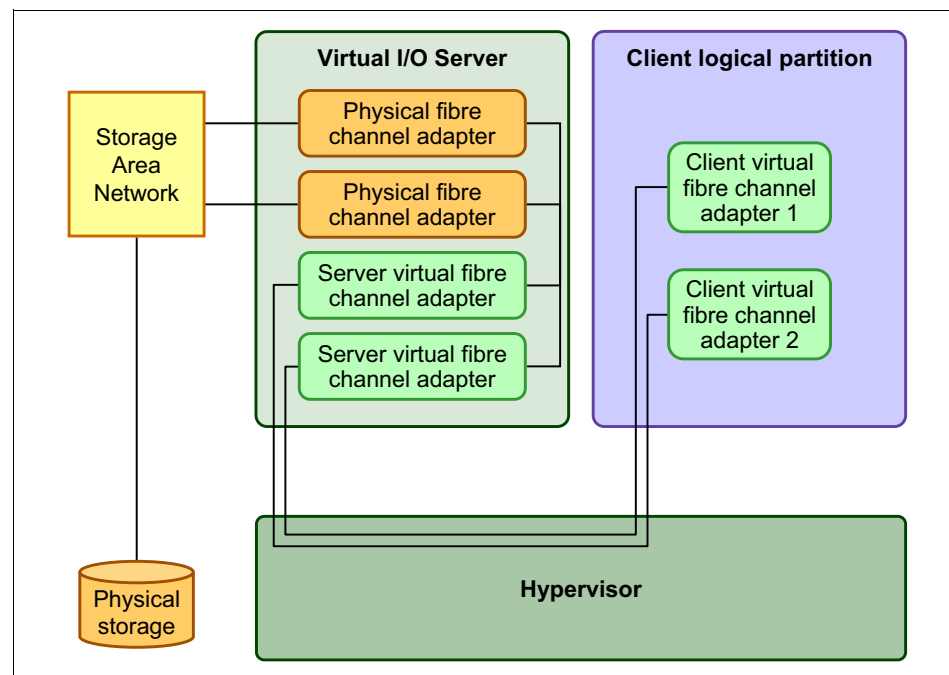


Figure 10-7 Host bus adapter failover

Figure 10-7 shows the following connections:

- ▶ The SAN connects physical storage to two physical Fibre Channel adapters located on the managed system.
- ▶ Two physical Fibre Channel adapters are assigned to the Virtual I/O Server partition and support Virtual Fibre Channel.
- ▶ The physical Fibre Channel ports are each connected to a virtual Fibre Channel server adapter on the Virtual I/O Server. The two virtual Fibre Channel server adapters on the Virtual I/O Server are connected to ports on two different physical Fibre Channel adapters to provide redundancy for the physical adapters.
- ▶ Each virtual Fibre Channel server adapter in the Virtual I/O Server partition is connected to one virtual Fibre Channel client adapter on a virtual I/O client partition. Each virtual Fibre Channel client adapter on each virtual I/O client partition receives a pair of unique WWPNs. The virtual I/O client partition uses one WWPN to log into the SAN at any given time. The other WWPN is used when the client logical partition is moved to another managed system using PowerVM Live Partition Mobility.

The virtual Fibre Channel adapters always have a one-to-one relationship between the virtual I/O client partitions and the virtual Fibre Channel adapters in the Virtual I/O Server partition. That is, each virtual Fibre Channel client adapter that is assigned to a virtual I/O client partition must connect to only one virtual Fibre Channel server adapter in the Virtual I/O Server partition, and each virtual Fibre Channel server adapter in the Virtual I/O Server partition must connect to only one virtual Fibre Channel client adapter in a virtual I/O client partition.

- ▶ Because multipathing is used in the virtual I/O client partition, it can access the physical storage through virtual Fibre Channel client adapter 1 or 2. If a physical Fibre Channel adapter in the Virtual I/O Server fails, the virtual I/O client uses the alternate path. This example does not show redundancy in the physical storage, but rather assumes it will be built into the SAN storage device.

Host bus adapter and Virtual I/O Server redundancy

A host bus adapter and Virtual I/O Server redundancy configuration provides a more advanced level of redundancy for the virtual I/O client partition.

Figure 10-8 shows an example of this redundancy configuration.

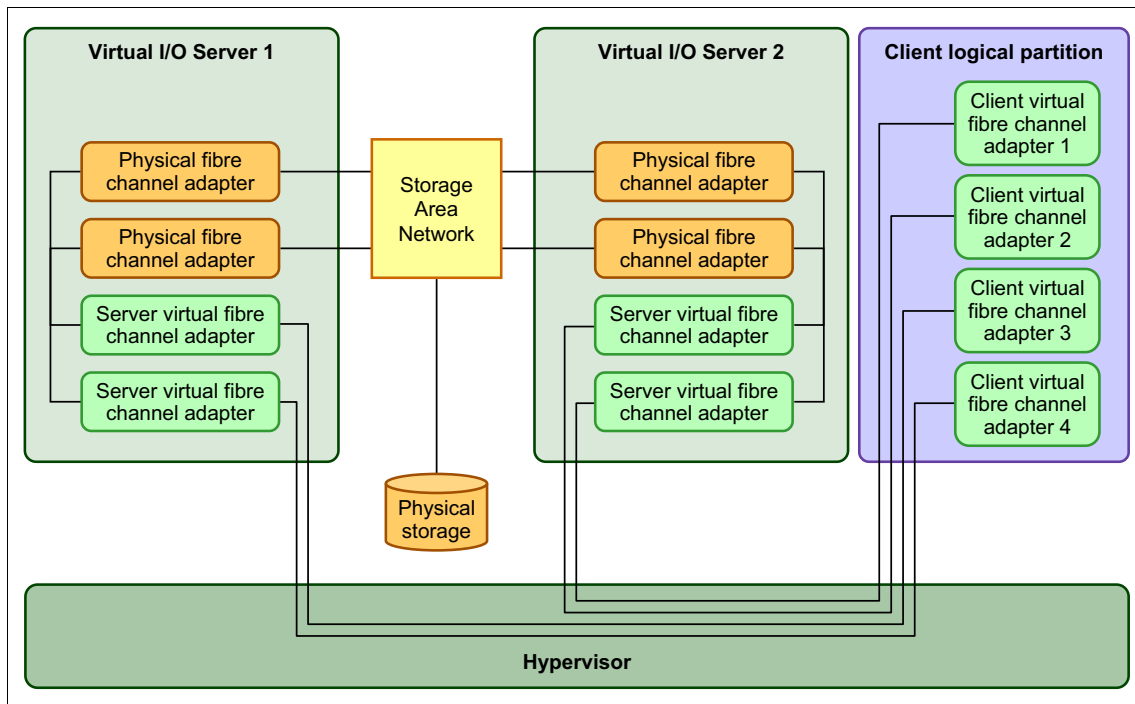


Figure 10-8 Host bus adapter and Virtual I/O Server failover

Figure 10-8 shows the following connections:

- ▶ The SAN connects physical storage to four physical Fibre Channel adapters located on the managed system.
- ▶ There are two Virtual I/O Server partitions to provide redundancy at the Virtual I/O Server level.
- ▶ Two physical Fibre Channel adapters are assigned to their respective Virtual I/O Server partitions and support Virtual Fibre Channel.
- ▶ The physical Fibre Channel ports are each connected to a virtual Fibre Channel server adapter on the Virtual I/O Server partition. The two virtual Fibre Channel server adapters on the Virtual I/O Server are connected to ports on two different physical Fibre Channel adapters to provide the most redundant solution for the physical adapters.
- ▶ Each virtual Fibre Channel server adapter in the Virtual I/O Server partition is connected to one virtual Fibre Channel client adapter in a virtual I/O client partition. Each virtual Fibre Channel client adapter on each virtual I/O client partition receives a pair of unique WWPNs. The client logical partition uses one WWPN to log into the SAN at any given time.

The other WWPN is used when the client logical partition is moved to another managed system by PowerVM Live Partition Mobility.

The virtual I/O client partition can access the physical storage through virtual Fibre Channel client adapter 1 or 2 on the client logical partition through Virtual I/O Server 2. The client can also access the physical storage through virtual Fibre Channel client adapter 3 or 4 on the client logical partition through Virtual I/O Server 1. If a physical Fibre Channel adapter fails on Virtual I/O Server 1, the client uses the other physical adapter connected to Virtual I/O Server 1 or uses the paths connected through Virtual I/O Server 2. If Virtual I/O Server 1 needs to be shut down for maintenance reasons, then the client uses the path through Virtual I/O Server 2. This example does not show redundancy in the physical storage, but rather assumes it will be built into the SAN.

Heterogeneous configuration with Virtual Fibre Channel

Combining virtual Fibre Channel client adapters with physical adapters in the client logical partition using AIX native MPIO or IBM i multipathing is supported, as shown in Figure 10-9. One virtual Fibre Channel client adapter and one physical adapter form two paths to the same LUN.

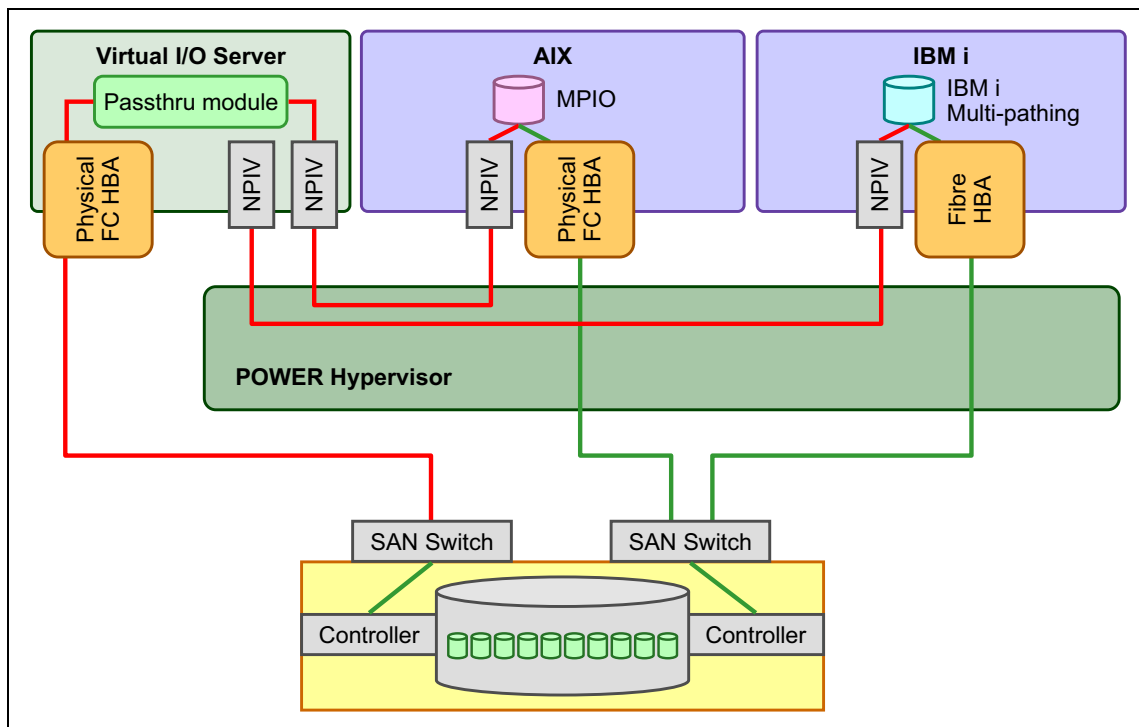


Figure 10-9 Heterogeneous multipathing configuration with Virtual Fibre Channel

Redundancy considerations for Virtual Fibre Channel

These examples can become more complex as you add physical storage redundancy and multiple clients, but the concepts remain the same. Consider the following points:

- ▶ To avoid configuring the physical Fibre Channel adapter to be a single point of failure for the connection between the virtual I/O client partition and its physical storage on the SAN, do not connect two virtual Fibre Channel client adapters from the same virtual I/O client partition to the same physical Fibre Channel adapter in the Virtual I/O Server partition. Instead, connect each virtual Fibre Channel server adapter to a different physical Fibre Channel adapter.
- ▶ Consider load balancing when mapping a virtual Fibre Channel server adapter in the Virtual I/O Server partition to a physical port on the physical Fibre Channel adapter.
- ▶ Consider what level of redundancy already exists in the SAN to determine whether to configure multiple physical storage units.
- ▶ Consider using two Virtual I/O Server partitions. Because the Virtual I/O Server is central to communication between virtual I/O client partitions and the external network, it is important to provide a level of redundancy for the Virtual I/O Server especially to prevent disruptions for maintenance actions such as a Virtual I/O Server upgrade requiring a reboot for activation. Multiple Virtual I/O Server partitions require more resources as well, so plan accordingly.
- ▶ Virtual Fibre Channel technology is useful when you want to move logical partitions between servers. For example, in an active PowerVM Live Partition Mobility environment, if you use the redundant configurations previously described in combination with physical adapters, you can stop all I/O activity through the dedicated, physical adapter and direct all traffic through a virtual Fibre Channel client adapter until the virtual I/O client partition is successfully moved. The dedicated physical adapter needs to be connected to the same storage as the virtual path.

Because you cannot migrate a physical adapter, all I/O activity is routed through the virtual path while you move the partition. After the logical partition is moved successfully, you need to set up the dedicated path (on the destination virtual I/O client partition) if you want to use the same redundancy configuration as you had configured on the original logical partition. Then the I/O activity can resume through the dedicated adapter, using the virtual Fibre Channel client adapter as a secondary path.

Figure 10-10 shows a managed system configured to use Virtual Fibre Channel, running two Virtual I/O Server partitions each with one physical Fibre Channel card. Each Virtual I/O Server partition provides virtual Fibre Channel adapters to the virtual I/O client. For increased serviceability multipathing is used in the virtual I/O client partitions.

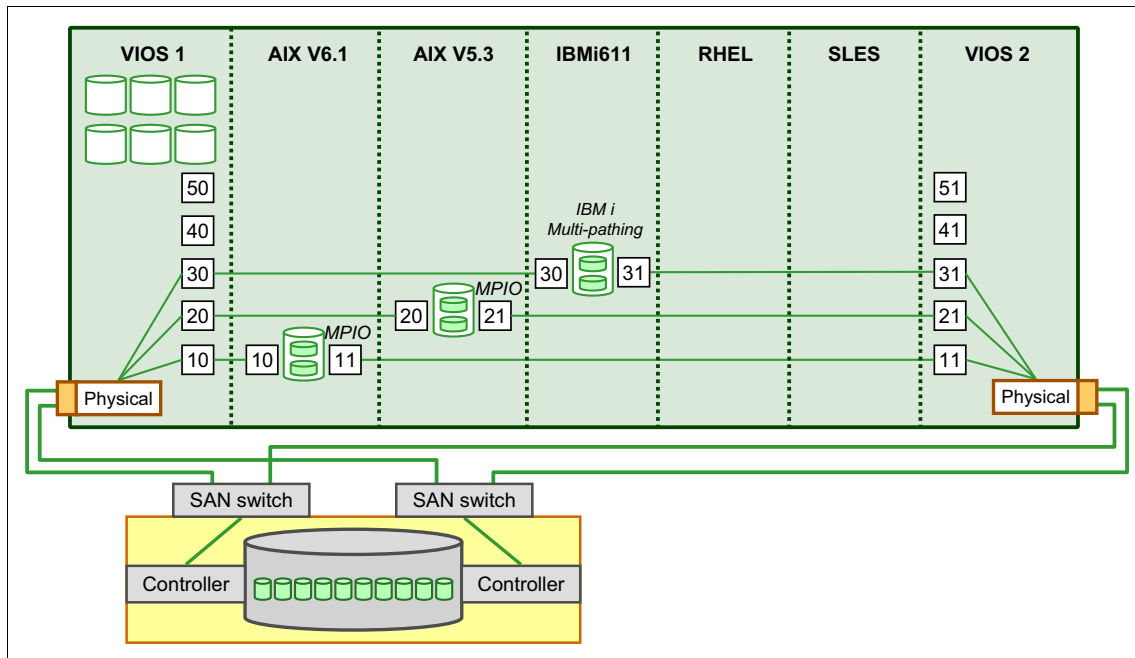


Figure 10-10 Redundant Virtual I/O Server partitions with Virtual Fibre Channel

Figure 10-10 shows the following connections:

- ▶ A SAN connects several LUNs from an external physical storage system to a physical Fibre Channel adapter that is located on the managed system. Each LUN is connected through both Virtual I/O Servers for redundancy. The physical Fibre Channel adapter is assigned to the Virtual I/O Server and supports Virtual Fibre Channel.
- ▶ There are five virtual Fibre Channel adapters available in each of the two Virtual I/O Servers. Three of them are mapped with the physical Fibre Channel adapter (adapter slots 10, 20 and 30 respectively 11, 21 and 31). All three virtual Fibre Channel server adapters are mapped to the same physical port on the physical Fibre Channel adapter.

- Each virtual Fibre Channel server adapter on the Virtual I/O Server partition connects to one virtual Fibre Channel client adapter on a virtual I/O client partition. Each virtual Fibre Channel client adapter receives a pair of unique WWPNs. The pair is critical, and both must be zoned if Live Partition Migration is planned to be used for AIX or Linux. The virtual I/O client partition uses one WWPN to log into the SAN at any given time. The other WWPN is used by the system when you move the virtual I/O client partition to another managed system with PowerVM Live Partition Mobility.

Using their unique WWPNs and the virtual Fibre Channel connections to the physical Fibre Channel adapter, the client operating system that runs in the virtual I/O client partitions discovers, instantiates, and manages the physical storage located on the SAN as if it were natively connected to the SAN storage device. The Virtual I/O Server provides the virtual I/O client partitions with a connection to the physical Fibre Channel adapters on the managed system.

There is always a one-to-one relationship between the virtual Fibre Channel client adapter and the virtual Fibre Channel server adapter.

Using the SAN tools of the SAN switch vendor, you can zone your Virtual Fibre Channel-enabled switch to include WWPNs that are created by the HMC for any virtual Fibre Channel client adapter on virtual I/O client partitions with the WWPNs from your storage device in a zone, as for a physical environment. The SAN uses zones to provide access to the targets based on WWPNs.

Redundancy configurations help to increase the serviceability of your Virtual I/O Server environment. With virtual Fibre Channel, you can configure the managed system so that multiple virtual I/O client partitions can independently access physical storage through the same physical Fibre Channel adapter. Each virtual Fibre Channel client adapter is identified by a unique WWPN, which means that you can connect each virtual I/O partition to independent physical storage on a SAN.

Similar to virtual SCSI redundancy, virtual Fibre Channel redundancy can be achieved using multipathing or mirroring at the virtual I/O client partition. The difference between traditional redundancy with SCSI adapters and the Virtual Fibre Channel technology using virtual Fibre Channel adapters is that the redundancy occurs on the client, because only the client recognizes the disk. The Virtual I/O Server is essentially just a pass-through managing the data transfer through the POWER hypervisor.

Mixtures: Though any mixture of Virtual I/O Server native SCSI, virtual SCSI, and virtual Fibre Channel I/O traffic is supported on the same physical FC adapter port, consider the implications that this might have for the manageability and serviceability of such a mixed configuration.

10.2.4 Virtual SCSI and Virtual Fibre Channel comparison

Virtual SCSI and Virtual Fibre Channel both offer significant benefits by enabling shared utilization of physical I/O resources. In the following paragraphs, we compare both capabilities and provide guidance for selecting the most suitable option.

Overview

Table 10-3 shows a high level comparison of virtual SCSI and Virtual Fibre Channel

Table 10-3 Virtual SCSI and Virtual Fibre Channel comparison

Feature	Virtual SCSI	Virtual Fibre Channel
Server based storage virtualization	Yes	No
Adapter level sharing	Yes	Yes
Device level sharing	Yes	No
LPM, AMS, Suspend Resume capable	Yes	Yes
Shared storage pool capable	Yes	No
SCSI-3 compliant (persistent reserve)	No ^a	Yes
Generic device interface	Yes	No
Tape library and LANfree backup support	No	Yes
Virtual tape and virtual optical support	Yes	No
Support for IBM PowerHA System Mirror for i ^b	No	Yes

a. Unless using Shared Storage Pools

b. Only applies to IBM i partitions

Components and features

In this section we describe the various components and features.

Device types

Virtual SCSI provides virtualized access to disk devices, optical devices, and tape devices.

With Virtual Fibre Channel, SAN disk devices and tape libraries can be attached. The access to tape libraries enables the use of LAN-Free backup, which is not possible with virtual SCSI.

Adapter and device sharing

Virtual SCSI allows sharing of physical storage adapters. It also allows sharing of storage devices by creating storage pools that can be partitioned to provide logical volume or file backed devices.

Virtual Fibre Channel technology allows sharing of physical Fibre Channel adapters only.

Hardware requirements

Virtual Fibre Channel implementation requires Virtual Fibre Channel capable Fibre Channel adapters on the Virtual I/O Server as well as Virtual Fibre Channel capable SAN switches.

Virtual SCSI supports a broad range of physical adapters.

Storage virtualization

Virtual SCSI server provides servers based storage virtualization. Storage resources can be aggregated and pooled on the Virtual I/O Server.

When using Virtual Fibre Channel, the Virtual I/O Server is only passing-through I/O to the client partition. Storage virtualization is done on the storage infrastructure in the SAN.

Storage assignment

With virtual SCSI, the storage is assigned (zoned) to the Virtual I/O Servers. From a storage administration perspective, there is no end-to-end view to see which storage is allocated to which client partition. When new disks are added to an existing client partition, they have to be mapped accordingly on the Virtual I/O Server. When using Live Partition Mobility storage needs to be assigned to the Virtual I/O Servers on the target server.

With Virtual Fibre Channel, the storage is assigned to the client partitions, as in an environment where physical adapters are used. No intervention is required on the Virtual I/O Server when new disks are added to an existing partition. When using Live Partition Mobility, storage moves to the target server without requiring a reassignment because the virtual Fibre Channels have their own WWPNs that move with the client partitions to the target server.

Support of PowerVM capabilities

Both virtual SCSI and Virtual Fibre Channel support most PowerVM capabilities such as Live Partition Mobility, Suspend and Resume, or Active Memory Sharing.

Virtual Fibre Channel does not support virtualization capabilities that are based on the shared storage pool such as thin provisioning.

Client partition considerations

Virtual SCSI uses a generic device interface. That means regardless of the backing device used the devices appear in the same way in the client partition. When using virtual SCSI, no additional device drivers need to be installed in the client partition. Virtual SCSI does not support load balancing across virtual adapters in a client partition.

With Virtual Fibre Channel, device drivers such as SDD, SDDPCM, or Atape need to be installed in the client partition for the disk devices or tape devices. SDD or SDDPCM allow load balancing across virtual adapters. Upgrading of these drivers requires special attention when you are using SAN devices as boot disks for the operating system.

World Wide Port Names

With redundant configurations using two Virtual I/O Servers and two physical Fibre Channel adapters as shown in 10.2.3, “Redundancy configurations for virtual Fibre Channel adapters” on page 193, up to 8 World Wide Port Names (WWPNs) will be used. Some SAN storage devices have a limit on the number of WWPNs they can manage. Therefore, before deploying Virtual Fibre Channel, verify that the SAN infrastructure can support the planned number of WWPNs.

Virtual SCSI uses only WWPNs of the physical adapters on the Virtual I/O Server.

Hybrid configurations

Virtual SCSI and Virtual Fibre Channel can be deployed in hybrid configurations. The next two examples show how both capabilities can be combined in real-world scenarios:

1. In an environment constrained in the number of WWPNs, virtual SCSI can be used to provide access to disk devices.
2. Partitions that require LAN-Free backup, access to tape libraries can be provided using Virtual Fibre Channel.

To simplify the upgrade of device drivers, Virtual Fibre Channel can be used to provide access to application data, while virtual SCSI can be used for access to the operating system boot disks.

10.2.5 Virtual optical devices

For more information on virtual optical devices, see the setup part of this book in 16.2.3, “Virtual optical” on page 490.

10.2.6 Virtual tape devices

For more information on virtual tape devices, see the setup part of this book in 16.2.4, “Virtual tape” on page 492.

10.2.7 Availability planning for virtual storage

This section gives you planning details required to set up redundancy for virtual storage.

Virtual storage redundancy

Virtual Fibre Channel or virtual SCSI redundancy can be achieved using MPIO and LVM mirroring at the client partition and Virtual I/O Server level.

Figure 10-8 on page 195 depicts a redundant virtual Fibre Channel configuration. See that section to understand how to implement highly reliable virtual I/O storage configurations based on Virtual Fibre Channel technology.

Figure 10-11 next depicts a virtual SCSI redundancy advanced setup using both MPIO and LVM mirroring in the client partition at the same time. Two Virtual I/O Servers host disks for a client partition. The client is using MPIO to access a SAN disk and LVM mirroring to access two SCSI disks. From the client perspective, the following situations can be handled without causing downtime for the client:

- ▶ Either path to the SAN disk can fail, but the client will still be able to access the data on the SAN disk through the other path. No action has to be taken to reintegrate the failed path to the SAN disk after repair if MPIO is configured as described in 16.2.6, “Availability configurations using multipathing” on page 501.
- ▶ The failure of a SCSI disk will cause stale partitions on AIX for the volume group with the assigned virtual disks, a suspended disk units on IBM i, or a disk marked as failed on Linux. The client partition will still be able to access the data on the second copy of the mirrored disk. After the failed disk is available again, the stale partitions have to be synchronized on the AIX client using the **varyonvg** command, the IBM i client will automatically resume mirrored protection, while on the Linux client, the command **mdadm** and a rescan of the devices will be required. These scenarios are described in 16.2.7, “Availability configurations using mirroring” on page 534.
- ▶ Either Virtual I/O Server can be rebooted for maintenance. This will result in a temporary simultaneous failure of one path to the SAN disk and stale partitions for the volume group on the SCSI disks, as described before.

Figure 10-11 illustrates these concepts.

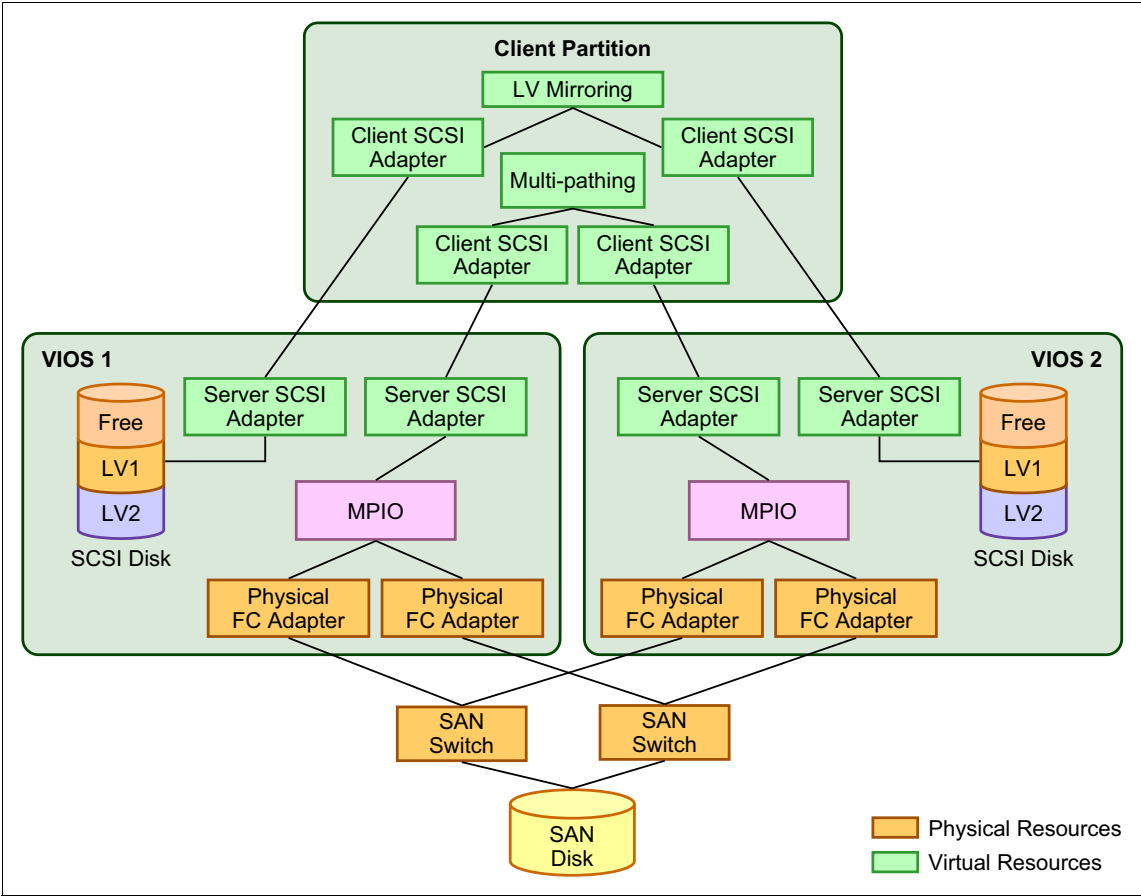


Figure 10-11 Virtual SCSI redundancy using multipathing and mirroring

Considerations for redundancy:

- ▶ If mirroring and multipathing are both configurable in your setup, multipathing is the preferred method for adding disk connection redundancy to the client. Mirroring causes stale partitions on AIX or Linux, and suspended disk units on IBM i, which require synchronization, whereas multipathing does not. Depending on the RAID level used on the SAN disks, the disk space requirements for mirroring can be higher, though mirroring across two storage systems even allows enhancement of the redundancy provided in a single storage system by RAID technology.
- ▶ Two Fibre Channel adapters in each Virtual I/O Server allow for adapter redundancy.

For further examples of virtual SCSI configurations, see 10.2.12, “Supported virtual SCSI configurations” on page 209.

The following sections describe using mirroring for each different AIX, IBM i, and Linux client partition across two Virtual I/O Servers.

10.2.8 AIX LVM mirroring in the client partition

In order to provide storage redundancy in the AIX client partition, AIX LVM mirroring can be used for virtual Fibre Channel devices or virtual SCSI devices.

When using virtual SCSI and AIX client partition mirroring between two storage subsystems, in certain situations, errors on hdisks located on a single storage subsystem can cause all hdisks connected to a virtual SCSI adapter to become inaccessible.

To avoid losing access to mirrored data, the best approach is to provide the disks of each mirror copy through a different virtual SCSI adapter as in Figure 10-12.

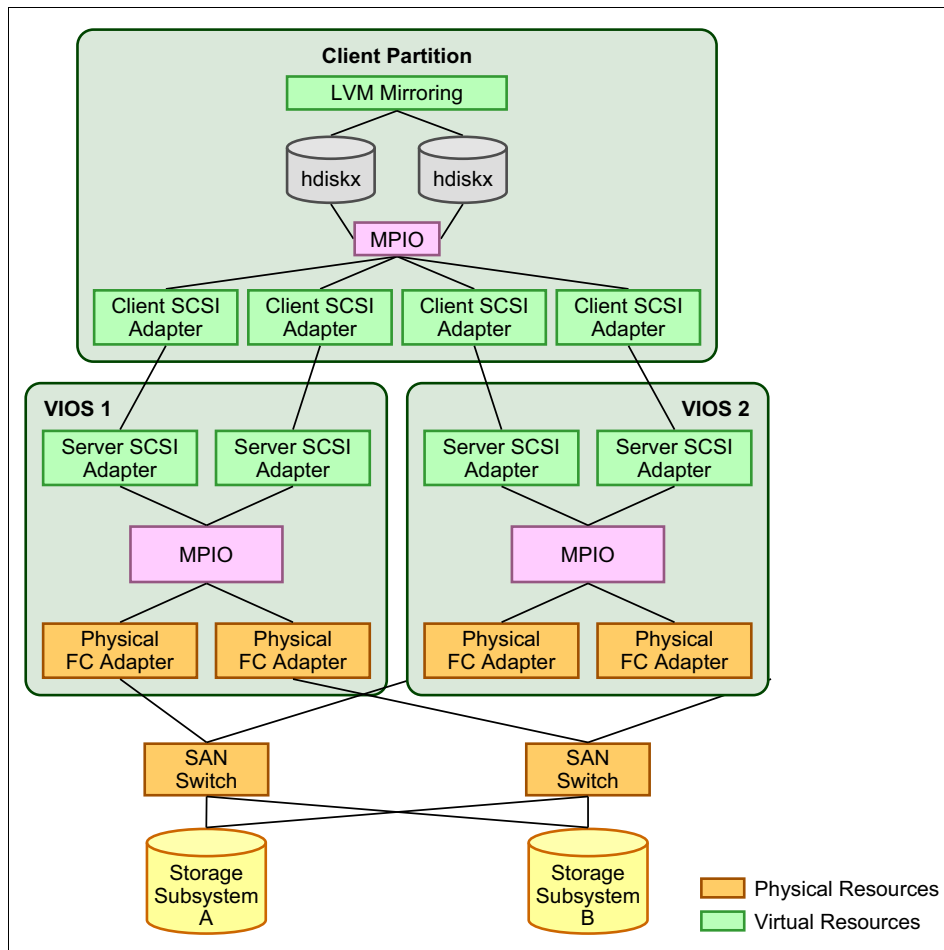


Figure 10-12 LVM mirroring with two storage subsystems

AIX client volume group mirroring is also required when a Virtual I/O Server logical volume is used as a virtual SCSI device on the client. In this case, the virtual SCSI devices are associated with different SCSI disks, each controlled by one of the two Virtual I/O Servers. You can find a sample LVM mirroring configuration in 16.2.7, “Availability configurations using mirroring” on page 534.

10.2.9 IBM i mirroring in the client partition

Using IBM i mirroring in the client partition to enable storage redundancy, ideally across two Virtual I/O Servers and two separate storage systems, is supported for virtual SCSI or DS8000 virtual Fibre Channel LUNs attached by Virtual Fibre Channel.

Virtual SCSI LUNs are always presented by the Virtual I/O Server as *unprotected* LUNs of type-model 6B22-050 to the IBM i client so they are always eligible for IBM i mirroring. For DS8000 virtual Fibre Channel LUNs, as with DS8000 native attachment, the LUNs need to be created as *unprotected* models (OS/400 model A8x) on the DS8000 to be eligible for IBM i mirroring.

Important: Currently all virtual SCSI or Fibre Channel adapters report in on IBM i under the same bus number 255, which allows for *IOP-level* mirrored protection only. To implement the concept of *bus-level* mirrored protection for virtual LUNs with larger configurations having more than one virtual IOP per mirror side, in order not to compromise redundancy, consider iteratively adding LUNs from one IOP pair at a time to the auxiliary storage pool by selecting the LUNs from one virtual IOP from each mirror side.

See “IBM i client mirroring” on page 544 for an example of an IBM i mirroring configuration.

10.2.10 Linux mirroring in the client partition

Mirroring on Linux partitions is implemented with Linux software RAID functionality provided by *md* (Multiple Devices) device driver. The *md* driver combines devices in one array for performance improvements and redundancy.

An *md* device with RAID1 level indicates a mirrored device with redundancy. RAID devices on Linux are usually represented as *md0*, *md1*, and so on.

Linux software RAID devices are managed and listed with the **mdadm** command. You can also list RAID devices with the **cat /proc/mdstat** command.

All devices in a RAID1 array must have the same size, otherwise the smallest device space is used and any extra space on other devices is wasted.

See “Linux client mirroring” on page 568 for an example of a Linux mirroring configuration.

10.2.11 Supported AIX client configurations

This section discusses various supported configurations in a Virtual I/O Server environment. The following configurations are described:

- ▶ Supported virtual SCSI configurations with:
 - Mirrored virtual SCSI devices on the client and server
 - Multipath configurations in a SAN environment

The configurations described in this section are not a complete list of all available supported configurations. We show a collection of the most frequently adopted configurations that meet the requirements of most production environments.

For the latest information about Virtual I/O Server supported environments, visit the following website:

<http://www14.software.ibm.com/support/customer/sas/f/vios/home.html>

10.2.12 Supported virtual SCSI configurations

In this section we present a detailed discussion of virtual SCSI configurations, including supported and best configurations when using the Virtual I/O Server.

Supported configurations with mirrored VSCSI devices

Figure 10-13 shows a supported way for mirroring disks with only one Virtual I/O Server.

On the Virtual I/O Server, you either configure two logical volumes and map them to the vhost adapter assigned to the client partition or you directly map the hdisks to the vhost adapter. On the client partition, the mapped devices will appear as two disks. The AIX client mirrors the two virtual disks using standard AIX LVM mirroring.

In Figure 10-13, two separate physical disks on the Virtual I/O Server are attached to two separate physical adapters for high availability. This also works when the disks are attached to only one physical adapter, but in this case it does not protect against an adapter failure.

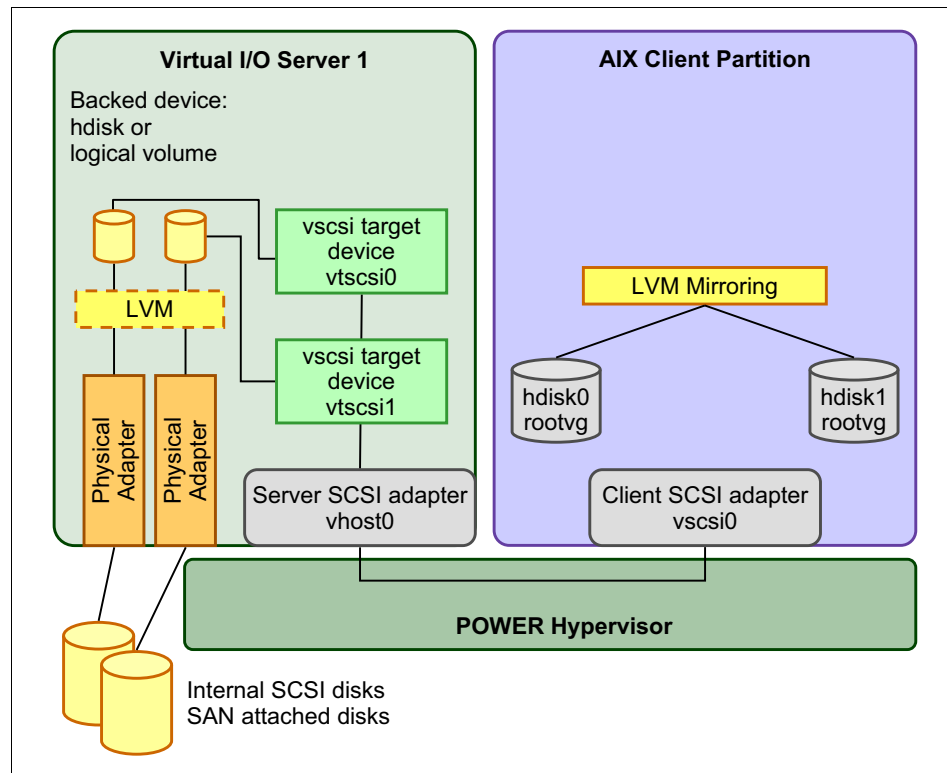


Figure 10-13 Supported and best ways to mirror virtual disks

Important: A logical volume of the Virtual I/O Server used as a virtual disk must not span multiple disks.

You can verify that a logical volume is confined to a single disk with the **lslv -pv lvname** command. The output of this command must only display a single disk.

Using mirroring, striping, or concatenation of physical disks using the LVM in the Virtual I/O client, or using such features of special RAID-capable host bus adapters or storage subsystems on the Virtual I/O Server, is best. Thus, to provide redundancy for the backed disk, a hardware RAID 5 array on the Virtual I/O Server can be used. Figure 10-14 shows the Virtual I/O Server configured with a SCSI RAID adapter.

A list of supported adapters can be found at this website:

<http://www14.software.ibm.com/webapp/set2/sas/f/vios/documentation/data sheet.html>

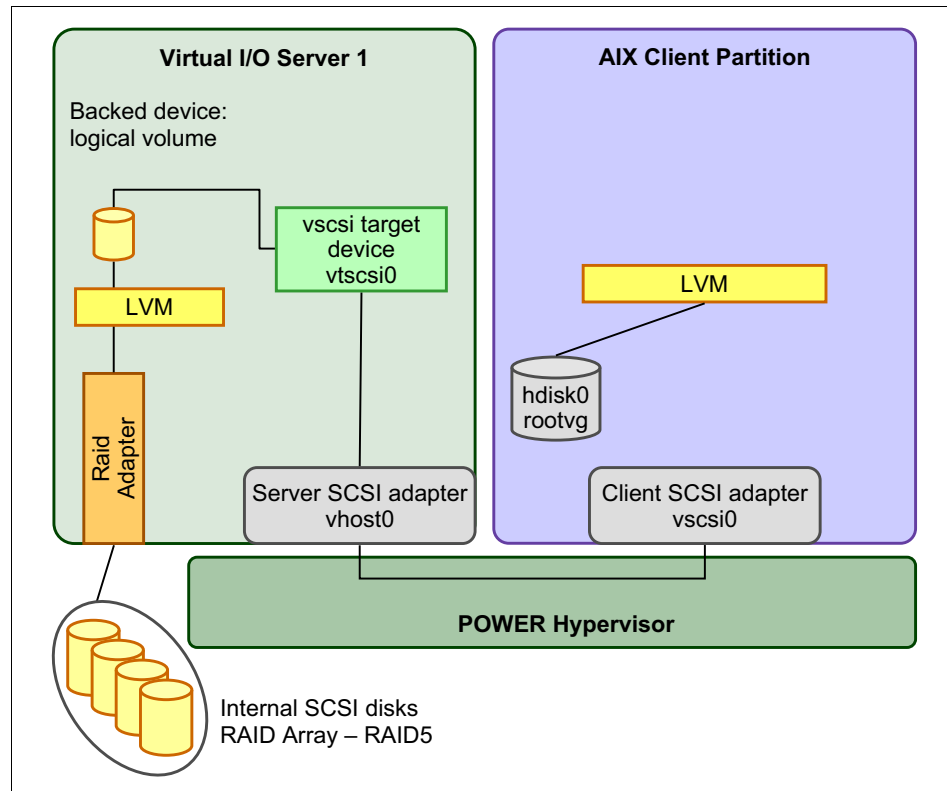


Figure 10-14 RAID5 configuration using a RAID adapter on the Virtual I/O Server

Attention: Only RAID hardware is supported for this configuration.

After creating a RAID 5 array, it will appear as one hdisk on the Virtual I/O Server. You can then divide the large disk into logical volumes and map them to your client partitions. Note that in this configuration, the RAID adapter is a single point of failure.

When using this configuration, plan for two additional disks for the installation of the Virtual I/O Server, which must be mirrored over two physical disks.

Important: Do not use the Virtual I/O Server rootvg for logical volumes that will be exported as virtual disks for the clients.

When planning for multiple Virtual I/O Servers, consider the best way for mirroring the virtual disks on the client partitions (Figure 10-15).

Support: One Virtual I/O Server is supported on the Integrated Virtualization Manager (IVM). Use the HMC to manage more than one Virtual I/O Server.

IBM supports up to ten Virtual I/O Servers within a single IBM Power System managed by an HMC. Though architecturally up to 254 partitions depending on IBM Power Systems models are supported, more than ten Virtual I/O Server partitions within a single server have not been tested and therefore are not advisable.

Either configure a logical volume on each Virtual I/O Server and map it to the vhost adapter that is assigned to the same client partition or directly map an hdisk to the appropriate vhost adapter. The mapped logical volume or hdisk will be configured as an hdisk on the client side, each belonging to a different Virtual I/O Server. Then use LVM mirroring on the client side.

Figure 10-15 shows the best way for mirroring the virtual disks on the client partitions.

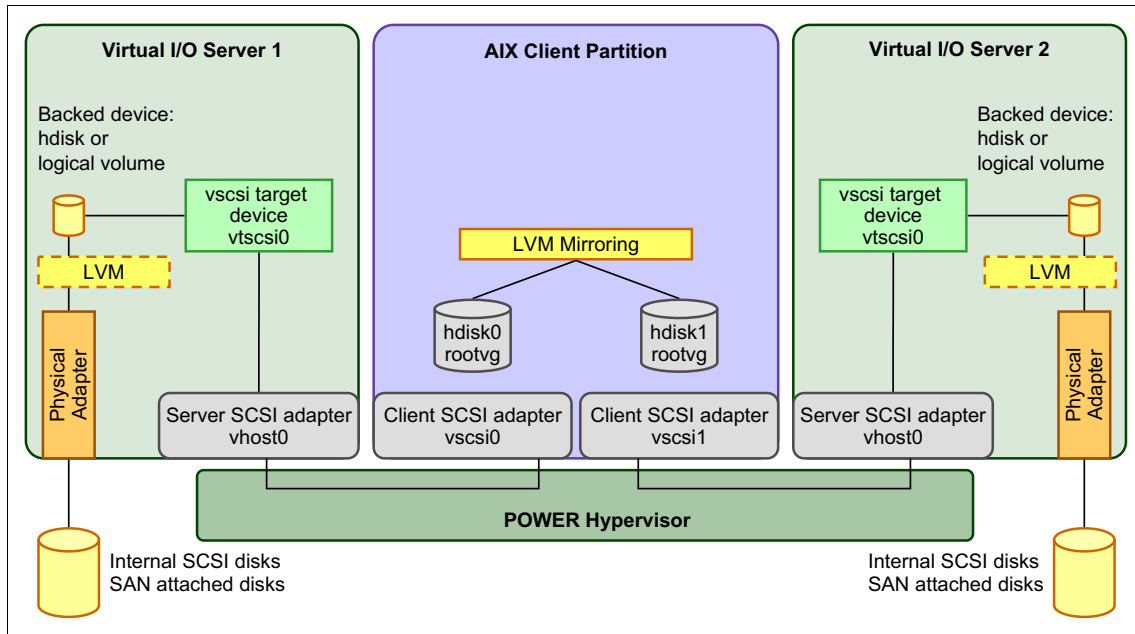


Figure 10-15 Best way to mirror virtual disks with two Virtual I/O Server

Supported multipath configurations in a SAN environment

When considering a supported configuration with MPIO, you need to distinguish two different scenarios:

- ▶ One Virtual I/O Server attaching a LUN in the SAN over more than one path. In this case, you only need to implement multipath software on the Virtual I/O Server.
- ▶ Having multiple Virtual I/O Servers connect to the same LUN and backed up to the same client. In this case, the client partition uses MPIO to access the virtual disk as a single device. You can also consider using multipath software on the Virtual I/O Server to access the LUN over more than one path for path failover and load balancing.

Support: This book only covers IBM storage solutions. For support statements of other storage vendors, contact your IBM representative or your storage vendor directly and ask for specifications on supported configurations.

Supported scenarios using one Virtual I/O Server

Next, Figure 10-16 shows a configuration with MPIO SDDPCM with only one Virtual I/O Server attached to IBM System Storage DS8000. The LUN is connected over two Fibre Channel adapters to the Virtual I/O Server to increase redundancy or throughput.

Because the disk is only attached to one Virtual I/O Server, it is possible to create logical volumes and map them to the vhost adapter that is assigned to the appropriate client partition. You can also choose to map the disk directly to the vhost adapter. For attaching the IBM System Storage DS8000 to only one Virtual I/O Server, MPIO with the SDDPCM or SDD is supported.

Important: The best method to attach IBM System Storage DS8000 is MPIO SDDPCM. Virtual disk devices created with SDD prior to SDD version 1.6.2.3 are not virtual to physical device compatible and might require a migration effort in the future. See the following website for more information about device compatibility in a Virtual I/O Server environment:

http://publib.boulder.ibm.com/infocenter/powersys/v3r1m5/index.jsp?topic=/iphb1/iphb1_vios_device_compat.htm

Figure 10-16 shows the aforementioned configuration with MPIO SDDPCM with only one Virtual I/O Server attached to IBM System Storage DS8000.

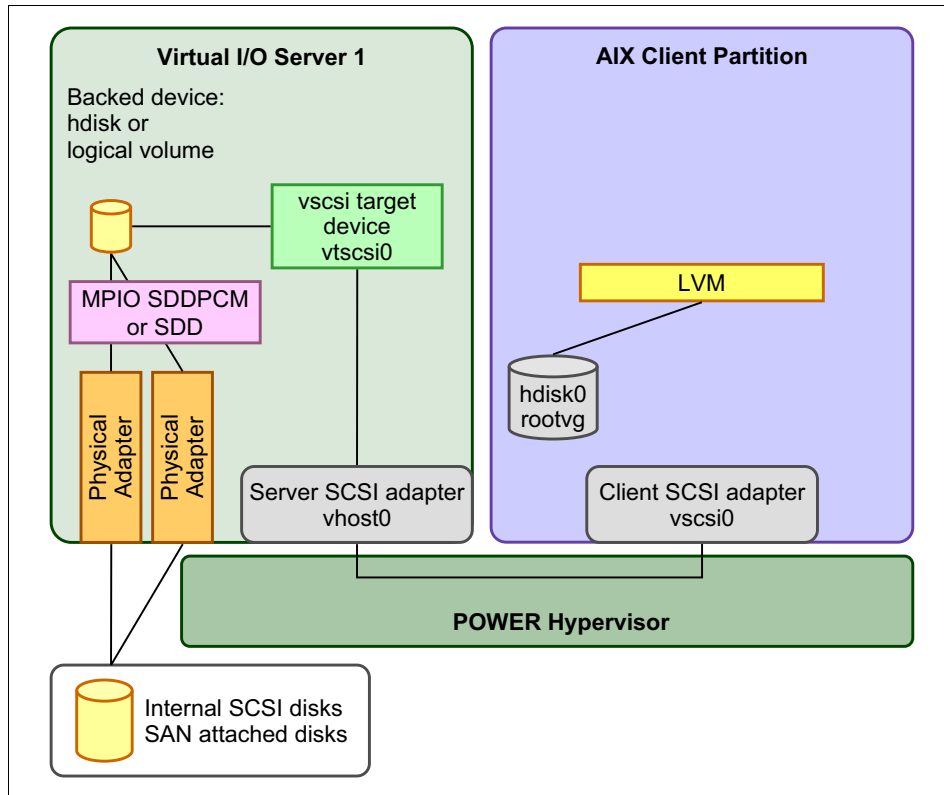


Figure 10-16 Using MPIO with IBM System Storage DS8000

Figure 10-17 shows the configuration with only one Virtual I/O Server for IBM TotalStorage DS Family using default MPIO.

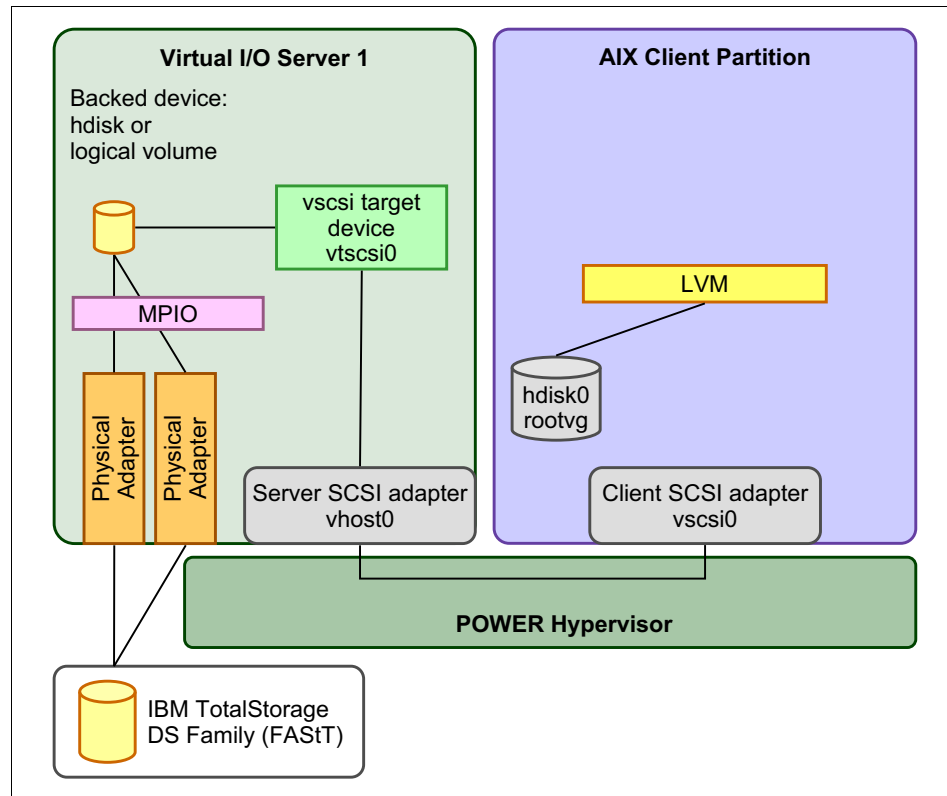


Figure 10-17 Using MPIO on the Virtual I/O Server with IBM TotalStorage

Supported scenarios using multiple Virtual I/O Servers

When configuring multiple Virtual I/O Servers, Figure 10-18 shows a supported configuration when attaching IBM TotalStorage SAN Volume Controller using multipath software on the Virtual I/O Server for additional redundancy and throughput.

Support: When using multiple Virtual I/O Servers, and exporting the same LUN to the client partitions, only mapping of hdisks is supported (logical volumes are not supported).

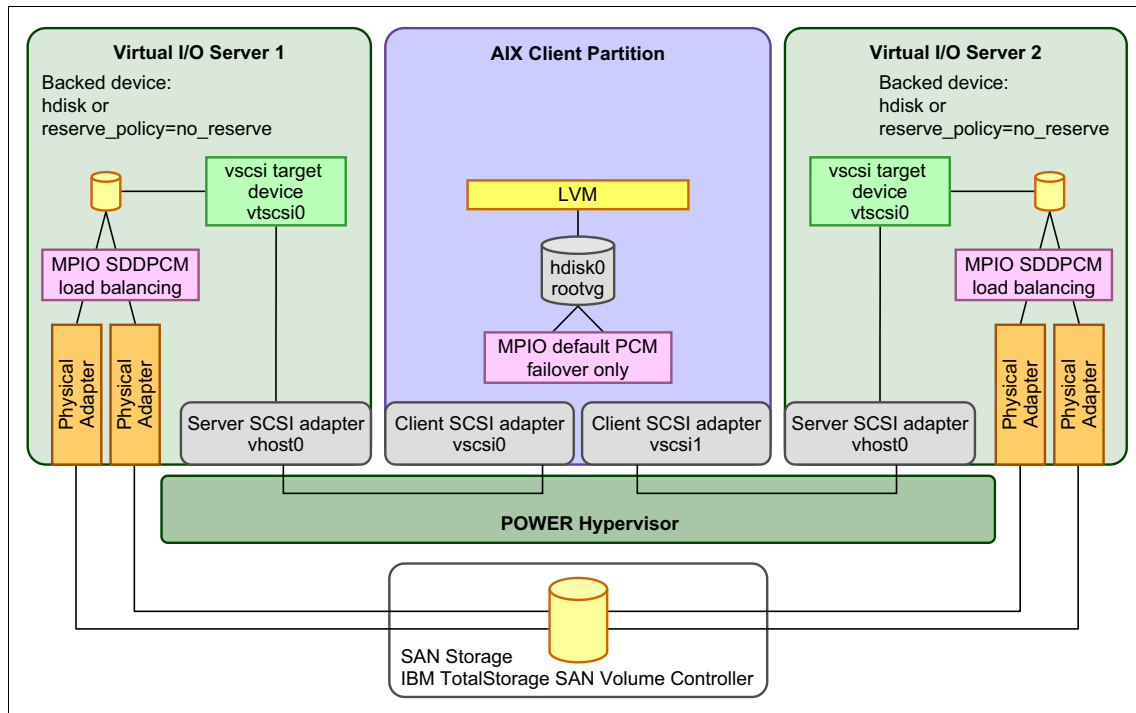


Figure 10-18 Configuration for IBM TotalStorage SAN Volume Controller

You can also attach IBM TotalStorage DS Family using the MPIO driver, as shown in Figure 10-19.

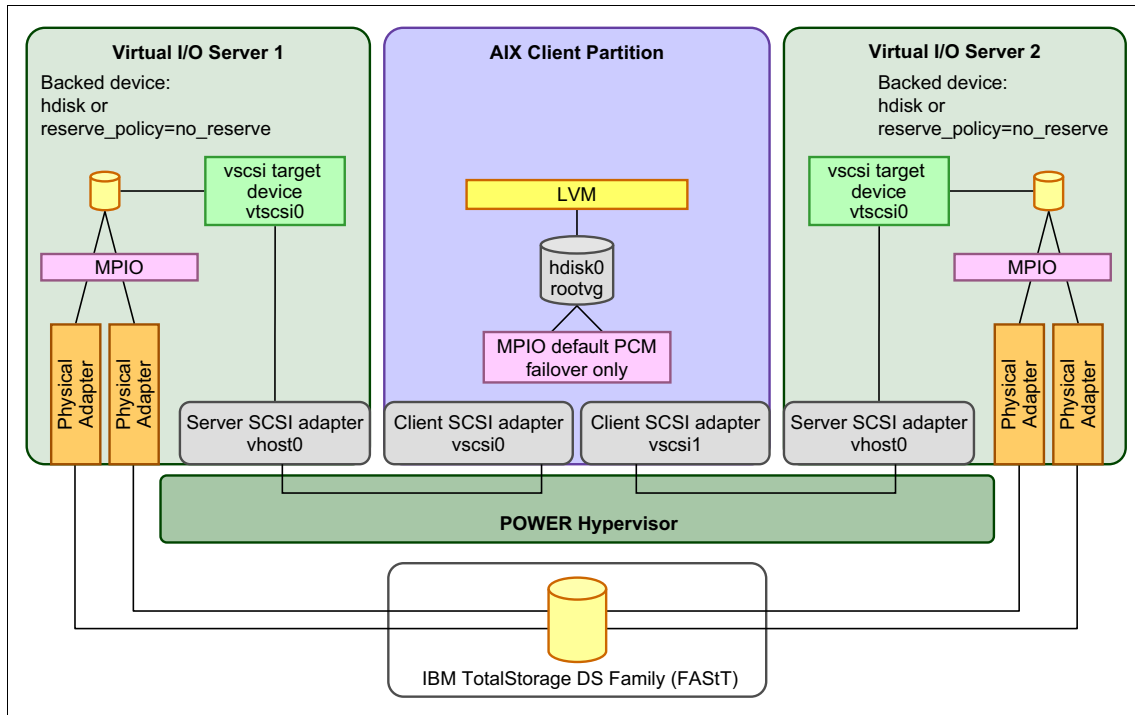


Figure 10-19 Configuration for multiple Virtual I/O Servers and IBM FASTT

On both Virtual I/O Servers, you have to set the hdisk attribute *reserve_policy* to no. This attribute prevents the Virtual I/O Server setting a reservation flag on the disk at the time of mapping. The MPIO component on the client partition will take the responsibility of managing the disk.

On the client partition MPIO, using the default PCM is supported, which only allows a failover policy and no load balancing. Only one path to the disks is active; the other path is used in the event that the active path fails, for example, when the Virtual I/O Server that serves the disk over the active path is rebooted.

It is possible to choose the active path on the client side. Users can manually configure the active paths for clients, enabling you to spread the workload evenly across the Virtual I/O Server. For detailed configuration steps, see “Availability configurations using multipathing” on page 501.

Reference: For the latest information about supported multipath drivers on storage subsystems, see the System Storage Interoperation Center (SSIC) website:

<http://www-03.ibm.com/systems/support/storage/config/ssic>

10.2.13 Shared storage pools

This section tells you about the necessary planning details for implementing shared storage pools in a PowerVM environment.

Following are the prerequisites for creating shared storage pools.

Prerequisites

Ensure that the following prerequisites are met:

- ▶ Virtual I/O Server version 2.2.1.3 Fix Pack 25, Service Pack 1 or newer.
- ▶ HMC version 7.7.4.0 or newer.
- ▶ The minimum storage required by your storage vendor

Configuring the Virtual I/O Server logical partitions

Configure the logical partitions with the following characteristics:

- ▶ There must be at least one CPU and one physical CPU of entitlement.
- ▶ The logical partitions must be configured as Virtual I/O Server logical partitions.
- ▶ The logical partitions must consists of at least 4 GB of memory.
- ▶ The logical partitions must consist of at least one physical Fibre Channel adapter.
- ▶ The rootvg device for a Virtual I/O Server logical partition cannot be included in storage pool provisioning.
- ▶ The Virtual I/O Server logical partitions in the cluster require access to all the SAN-based physical volumes in the shared storage pool of the cluster.
- ▶ One Virtual I/O Server logical partition must have a network connection either through an Integrated Virtual Ethernet adapter or through a physical adapter. On Virtual I/O Server version 2.2.2.0, clusters support virtual local area network (VLAN) tagging.

Scalability limits

Following are the scalability limits of Shared Storage Pool Cluster on Virtual I/O Server version 2.2.2.0:

- ▶ Max number of Nodes in cluster: 16
- ▶ Max Number of Physical Disks in Pool: 1024
- ▶ Max Number of Virtual Disks: 8192
- ▶ Max Number of Client LPARs per Virtual I/O Server: 125
- ▶ Max Capacity of Physical Disks in Pool: 4 TB
- ▶ Min/Max Storage Capacity of Storage Pool: 512 TB
- ▶ Max Capacity of a Virtual Disk (LU) in Pool: 4 TB

Restriction: You cannot use the logical units in a cluster as paging devices for PowerVM Active Memory Sharing or Suspend/Resume features.

Configuring client logical partitions

Configure the client partitions with the following characteristics:

- ▶ The client logical partitions must be configured as AIX or Linux client systems.
- ▶ They must have at least 1 GB of minimum memory.
- ▶ The associated rootvg device must be installed with the appropriate AIX or Linux system software.
- ▶ Each client logical partition must be configured with a sufficient number of virtual SCSI adapter connections to map to the virtual server SCSI adapter connections of the required Virtual I/O Server logical partitions.
- ▶ Virtual I/O Server 2.2.2.0 or later, clusters support virtual local area network (VLAN) tagging.

Network addressing considerations

Uninterrupted network connectivity is required for shared storage pool operations. The network interface that is used for the shared storage pool configuration must be on a highly reliable network, which is not congested.

Ensure that both the forward and reverse lookup for the hostname used by the Virtual I/O Server logical partition for clustering resolves to the same IP address.

Notes:

- ▶ Starting with Virtual I/O Server version 2.2.2.0, the SSP cluster can be created on an IPv6 configuration, Hence Virtual I/O Server logical partitions in a cluster can have hostnames that resolve to an IPv6 address. To set up an SSP cluster on an IPv6 network, IPv6 stateless autoconfiguration is suggested. You can have Virtual I/O Server logical partitions configured with either an IPv6 static configuration or an IPv6 stateless autoconfiguration. A Virtual I/O Server that has both IPv6 static configuration and IPv6 stateless autoconfiguration is not supported in Virtual I/O Server version 2.2.2.0.
- ▶ The hostname of each Virtual I/O Server logical partition that belongs to the same cluster must resolve to the same IP address family, which is either IPv4 or IPv6.
- ▶ To migrate to an IPv6 environment from an IPv4 environment, see *IBM PowerVM Virtualization Managing and Monitoring*, SG24-7590.

Notes:

- ▶ To change the hostname of a Virtual I/O Server logical partition in the cluster, you need to remove the partition from the cluster and change the hostname. Subsequently you can add the partition back to the cluster again with new hostname. More details can be found in *IBM PowerVM Virtualization Managing and Monitoring*, SG24-7590.
- ▶ With Virtual I/O Server version 2.2.2.0, or later, commands on Virtual I/O Server (**mktcpip**, **rmtcpip**, **chtcpip**, **hostmap**, **chdev**, **rmdev**) are enhanced to perform configuration of more than one network interface, without disturbing its existing network configuration. In the SSP environment, this feature helps the user to configure multiple network interfaces, without causing any harm to the existing SSP setup. In the presence of multiple network interfaces, the primary interface may not be the interface being used for cluster communication. In such an SSP environment, the user is not restricted from altering the network configuration of other interfaces.

Storage provisioning to Virtual I/O Server partitions

When a cluster is created, you must specify one physical volume for the repository physical volume and at least one physical volume for the storage pool physical volume. The storage pool physical volumes are used to provide storage to the actual data generated by the client partitions. The repository physical volume is used to perform cluster communication and store the cluster configuration. The maximum client storage capacity matches the total storage capacity of all storage pool physical volumes. The repository disk must have at least 10 GB of available storage space. The physical volumes in the storage pool must have at least 10 GB of available storage space in total.

Use any method that is available for the SAN vendor to create each physical volume with at least 10 GB of available storage space. Map the physical volume to the logical partition Fibre Channel adapter for each Virtual I/O Server in the cluster. The physical volumes must only be mapped to the Virtual I/O Server logical partitions connected to the shared storage pool.

Note: Each of the Virtual I/O Server logical partitions assigns hdisk names to all physical volumes available through the Fibre Channel ports, such as hdisk0 and hdisk1. The Virtual I/O Server logical partition might select different hdisk numbers for the same volumes to the other Virtual I/O Server logical partition in the same cluster. For example, the viosA1 Virtual I/O Server logical partition can have hdisk9 assigned to a specific SAN disk, whereas the viosA2 Virtual I/O Server logical partition can have the hdisk3 name assigned to that same disk. For some tasks, the unique device ID (UDID) can be used to distinguish the volumes. Use the **chkdev** command to obtain the UDID for each disk.

Set the Fibre Channel adapters parameters as follows:

```
chdev -dev fscsi0 -attr dyntrk=yes -perm  
chdev -dev fscsi0 -attr fc_err_recov=fast_fail -perm
```

The physical volumes need to have the reserve policy set to “**no_reserve**”. However, for repository physical volume, this is taken care of by the Cluster Aware AIX (CAA) layer on the Virtual I/O Server.

10.3 Network virtualization planning

The following sections help you to plan for network virtualization.

10.3.1 Virtual Ethernet

Virtual Ethernet technology facilitates IP-based communication between logical partitions on the same system using VLAN-capable software switch systems. Using Shared Ethernet Adapter technology, logical partitions can communicate with other systems outside the hardware unit without being assigned physical Ethernet slots.

You can create virtual Ethernet adapters using the Hardware Management Console (HMC) and configure them using the Virtual I/O Server command-line interface. You can also use the Integrated Virtualization Manager to create and manage virtual Ethernet adapters. With the Virtual I/O Server version 2.2 or later, you can add, remove, or modify the existing set of VLANs for a virtual Ethernet adapter that is assigned to an active partition on a POWER7 processor-based server by using the HMC. The server firmware level must be at least AH720_064 for high-end servers, AM720_064 for mid-range servers, and AL720_064 for low-end servers. The HMC must be at version 7.7.2.0, with mandatory eFix MH01235, or later, to perform this task.

Consider using virtual Ethernet on the Virtual I/O Server in the following situations:

- ▶ When using advanced PowerVM virtualization technologies like Live Partition Mobility or partition Suspend and Resume for which the client partitions are not allowed to have physical I/O devices assigned plan for using virtual Ethernet with a Shared Ethernet Adapter on the Virtual I/O Server.
- ▶ When the capacity or the bandwidth requirement of the individual logical partition is inconsistent with, or is less than, the total bandwidth of a physical Ethernet adapter. Logical partitions that use the full bandwidth or capacity of a physical Ethernet adapter should use dedicated Ethernet adapters.
- ▶ When you need an Ethernet connection, but there is no slot available in which to install a dedicated adapter.

10.3.2 Virtual LAN

In many situations, the physical network topology has to take into account the physical constraints of the environment, such as rooms, walls, floors, and buildings, to name a few.

On the other hand, VLANs can be independent of the physical topology. Figure 10-20 shows two VLANs (VLAN 1 and 2) defined on three switches (Switch A, B, and C). There are seven hosts (A-1, A-2, B-1, B-2, B-3, C-1, and C-2) connected to the three switches.

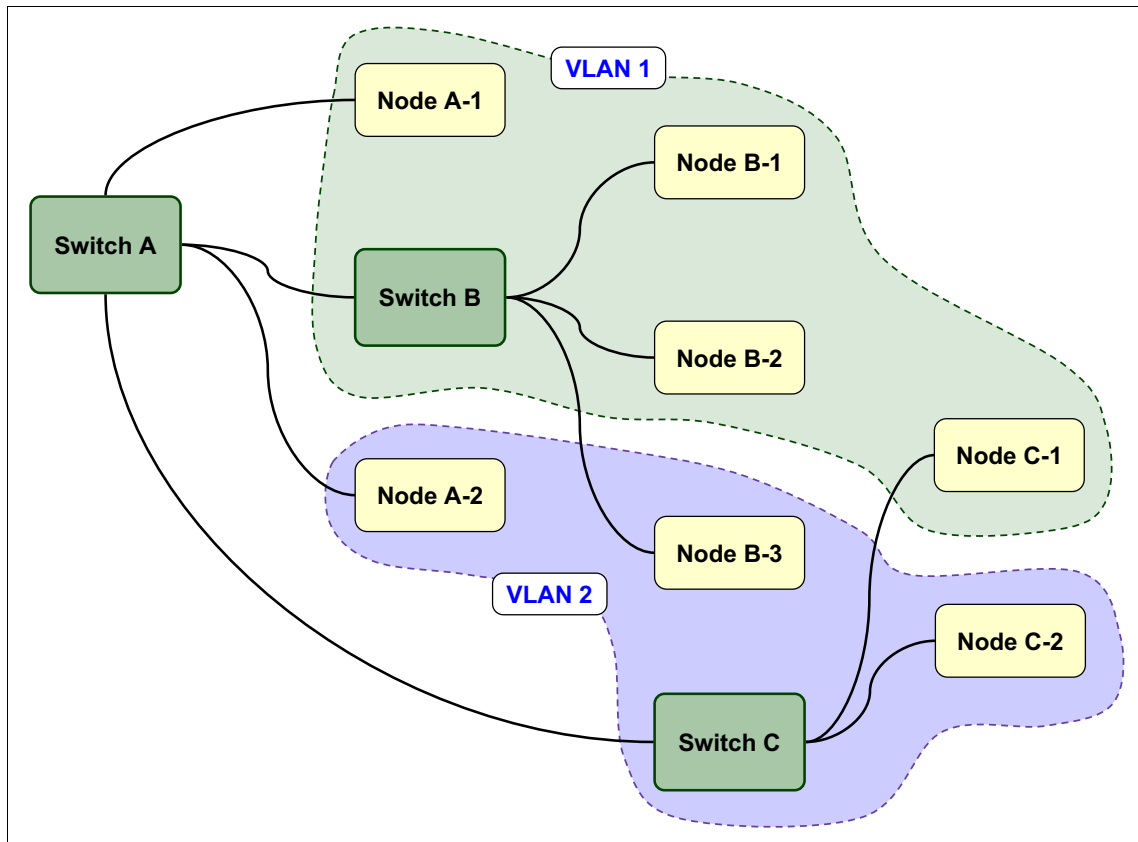


Figure 10-20 Example of VLANs

The physical network topology of the LAN forms a tree, which is typical for a non-redundant LAN:

- Switch A:
 - Node A-1
 - Node A-2
 - Switch B:
 - Node B-1
 - Node B-2
 - Node B-3

- Switch C:
 - Node C-1
 - Node C-2

Although nodes C-1 and C-2 are physically connected to the same switch C, traffic between two nodes is blocked:

- ▶ VLAN 1:
 - Node A-1
 - Node B-1
 - Node B-2
 - Node C-1
- ▶ VLAN 2:
 - Node A-2
 - Node B-3
 - Node C-2

To enable communication between VLAN 1 and 2, L3 routing or inter-VLAN bridging has to be established between them. The bridging is typically provided by an L3 device, for example, a router or firewall plugged into switch A.

Consider the uplinks between the switches: they carry traffic for both VLANs 1 and 2. Thus, there only has to be one physical uplink from B to A, not one per VLAN. The switches will not be confused and will not mix up the different VLANs' traffic, because packets travelling through the trunk ports over the uplink will have been tagged appropriately.

VLANs also have the potential to improve network performance. By splitting up a network into different VLANs, you also split up broadcast domains. Thus, when a node sends a broadcast, only the nodes on the same VLAN will be interrupted by receiving the broadcast. The reason is that normally broadcasts are not forwarded by routers. You have to keep this in mind if you implement VLANs and want to use protocols that rely on broadcasting, such as BOOTP or DHCP for IP autoconfiguration.

It is also common practice to use VLANs if Gigabit Ethernet's Jumbo Frames are implemented in an environment, where not all nodes or switches are able to use or are compatible with Jumbo Frames. Jumbo Frames allow for an MTU size of 9000 instead of Ethernet's default 1500. This can improve throughput and reduce processor load on the receiving node in a heavy loaded scenario, such as backing up files over the network.

VLANs can provide additional security by allowing an administrator to block packets from one domain to another domain on the same switch. This provides an additional control on what LAN traffic is visible to specific Ethernet ports on the switch. Packet filters and firewalls can be placed between VLANs, and Network Address Translation (NAT) can be implemented between VLANs. VLANs can make the system less vulnerable to attacks.

10.3.3 Virtual switches

The POWER Hypervisor switch is consistent with IEEE 802.1Q. It works on OSI-Layer 2 and supports up to 4094 networks (4094 VLAN IDs).

When a message arrives at a Logical LAN switch port from a Logical LAN adapter, the POWER Hypervisor caches the message's source MAC address to use as a filter for future messages to the adapter. The POWER Hypervisor then processes the message depending on whether the port is configured for IEEE VLAN headers. If the port is configured for VLAN headers, the VLAN header is checked against the port's allowable VLAN list. If the message specified VLAN is not in the port's configuration, the message is dropped. After the message passes the VLAN header check, it passes onto destination MAC address processing.

If the port is *not* configured for VLAN headers, the POWER Hypervisor inserts a two-byte VLAN header (based on the port's configured VLAN number) into the message. Next, the destination MAC address is processed by searching the table of cached MAC addresses.

If a match for the MAC address is not found and if no *trunk adapter* is defined for the specified VLAN number, the message is dropped; otherwise, if a match for the MAC address is not found and if a trunk adapter is defined for the specified VLAN number, the message is passed on to the trunk adapter. If a MAC address match is found, then the associated switch port's configured, allowable VLAN number table is scanned for a match to the VLAN number contained in the message's VLAN header. If a match is not found, the message is dropped.

Next, the VLAN header configuration of the destination switch port is checked. If the port is configured for VLAN headers, the message is delivered to the destination Logical LAN adapters, including any inserted VLAN header. If the port is configured for no VLAN headers, the VLAN header is removed before being delivered to the destination Logical LAN adapter.

Figure 10-21 shows a graphical representation of the behavior of the virtual Ethernet when processing packets.

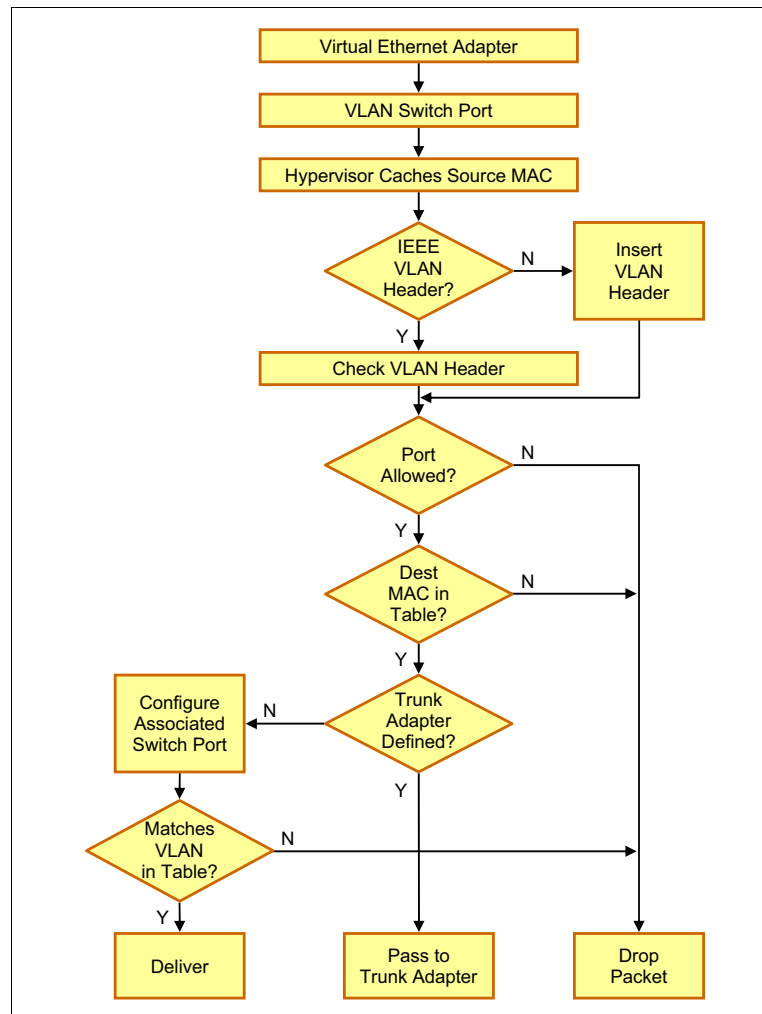


Figure 10-21 Flow chart of virtual Ethernet

Multiple virtual switches

POWER6 or later systems support multiple virtual switches. By default, a single virtual switch named “Ethernet0” is configured. This name can be changed dynamically and additional virtual switches can be created using a name of your choice.

Additional virtual switches can be used to provide an additional layer of security or to increase the flexibility of a virtual Ethernet configuration.

For example, to isolate traffic in a Demilitarized Zone (DMZ) from an internal network without relying entirely on VLAN separation, two virtual switches can be used. Systems that participate in the DMZ network will have their virtual adapters configured to use one virtual switch, whereas systems that participate in the internal network will be configured to use another.

Consider the following points when using multiple virtual switches:

- ▶ A virtual Ethernet adapter can only be associated with a single virtual switch.
- ▶ Each virtual switch supports the full range of VLAN IDs (1-4094).
- ▶ The same VLAN ID can exist in all virtual switches independently of each other.
- ▶ Virtual switches can be created and removed dynamically. However, a virtual switch cannot be removed if there is an active virtual Ethernet adapter using it.
- ▶ Virtual switch names can be modified dynamically without interruption to connected virtual Ethernet adapters.
- ▶ With Live Partition Mobility, virtual switch names must match between the source and target systems. The validation phase will fail if this is not true.
- ▶ All virtual adapters in a Shared Ethernet Adapter must be members of the same virtual switch.

Important: When using a Shared Ethernet Adapter, the name of the virtual switch is recorded in the configuration of the SEA on the Virtual I/O Server at creation time. If the virtual switch name is modified, the name change is not reflected in this configuration until the Virtual I/O Server is rebooted, or the SEA device is reconfigured. The `rmdev -l` command followed by `cfgmgr` is sufficient to update the configuration. If this is not updated, it can cause a Live Partition Migration validation process to fail because the Virtual I/O Server will still refer to old name.

ASM method versus HMC method: There is an alternate method for configuring virtual switches, accessible through the Advanced System Management (ASM) interface on the server as opposed to the HMC. Virtual switches configured by this interface do not behave in the same manner as HMC configured virtual switches. *In general, the preferred method is to use the HMC method.* Consult your IBM representative before attempting to modify the virtual switch configuration in the ASM.

10.3.4 Shared Ethernet Adapter

A Shared Ethernet Adapter (SEA) can be used to bridge a physical Ethernet network to a virtual Ethernet network. It also provides the ability for several client partitions to share one physical adapter. Using a SEA, you can connect internal and external VLANs using a physical adapter. The SEA hosted in the Virtual I/O Server acts as a layer-2 bridge between the internal and external network.

A SEA is a layer-2 network bridge to securely transport network traffic between virtual Ethernet networks and physical network adapters. The Shared Ethernet Adapter service runs in the Virtual I/O Server. It cannot be run in a general purpose AIX or Linux partition.

These are considerations regarding the use of SEA:

- ▶ Virtual Ethernet requires the POWER Hypervisor and PowerVM feature (Standard or Enterprise Edition) and the installation of a Virtual I/O Server.
- ▶ Virtual Ethernet cannot be used prior to AIX 5L Version 5.3. Thus, an AIX 5L Version 5.2 partition will need a physical Ethernet adapter.

Tip: A Linux partition can provide bridging function as well, with the **brctl** command.

The Shared Ethernet Adapter allows partitions to communicate outside the system without having to dedicate a physical I/O slot and a physical network adapter to a client partition. The Shared Ethernet Adapter has the following characteristics:

- ▶ Virtual Ethernet MAC addresses of virtual Ethernet adapters are visible to outside systems (using the **arp -a** command).
- ▶ Unicast, broadcast, and multicast is supported, so protocols that rely on broadcast or multicast, such as Address Resolution Protocol (ARP), Dynamic Host Configuration Protocol (DHCP), Boot Protocol (BOOTP), and Neighbor Discovery Protocol (NDP) can work across an SEA.

In order to bridge network traffic between the virtual Ethernet and external networks, the Virtual I/O Server has to be configured with at least one physical Ethernet adapter. One SEA can be shared by multiple virtual Ethernet adapters and each can support multiple VLANs.

Figure 10-22 shows a configuration example of an SEA with one physical and two virtual Ethernet adapters. A SEA can include up to 16 virtual Ethernet adapters on the Virtual I/O Server that share the physical access.

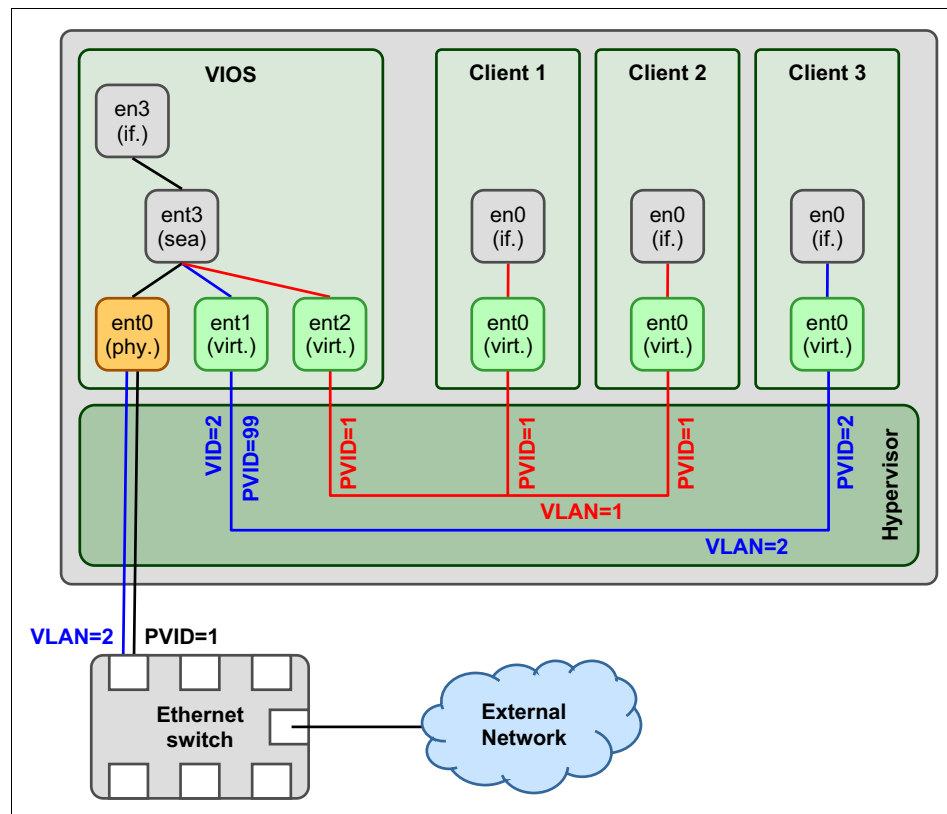


Figure 10-22 Shared Ethernet Adapter

Tip: A Shared Ethernet Adapter does not need to have IP configured to be able to perform the Ethernet bridging functionality. It is very convenient to configure IP on the Virtual I/O Server. This is because the Virtual I/O Server can then be reached by TCP/IP, for example, to perform dynamic LPAR operations or to enable remote login. This can be done by configuring an IP address directly on the SEA device, but it can also be defined on an additional virtual Ethernet adapter in the Virtual I/O Server carrying the IP address. This leaves the SEA without the IP address, allowing for maintenance on the SEA without losing IP connectivity if SEA failover has been configured. Neither has a remarkable impact on Ethernet performance.

When an Ethernet frame is sent from a virtual Ethernet adapter on a client partition to the POWER Hypervisor, the POWER Hypervisor searches for the destination MAC address within the VLAN. If no such MAC address exists within the VLAN, it forwards the frame to the virtual Ethernet adapter on the VLAN that has the Access External Networks option enabled. This virtual Ethernet adapter corresponds to a port of a layer-2 bridge, while the physical Ethernet adapter constitutes another port of the same bridge.

The SEA directs packets based on the VLAN ID tags. One of the virtual adapters in the Shared Ethernet Adapter on the Virtual I/O Server must be designated as the *default* PVID adapter. Ethernet frames without any VLAN ID tags that the SEA receives from the external network are forwarded to this adapter and assigned the default PVID. In Figure 10-22 on page 230, ent2 is designated as the default adapter, so all untagged frames received by ent0 from the external network will be forwarded to ent2. Because ent1 is not the default PVID adapter, only VID=2 will be used on this adapter, and the PVID=99 of ent1 is not important. It can be set to any unused VLAN ID. Alternatively, ent1 and ent2 can also be merged into a single virtual adapter ent1 with PVID=1 and VID=2, being flagged as the default adapter.

When the SEA receives or sends IP (IPv4 or IPv6) packets that are larger than the MTU of the adapter that the packet is forwarded through, either IP fragmentation is performed, or an ICMP packet too big message is returned to the sender, if the **Do not fragment** flag is set in the IP header. This is used, for example, with Path MTU discovery.

Theoretically, one adapter can act as the only contact with external networks for all client partitions. For more demanding network traffic scenarios (large number of client partitions or heavy network usage), it is important to adjust the throughput capabilities of the physical Ethernet configuration to accommodate the demand.

Note: Observe these considerations when implementing virtual adapters:

- ▶ Only Ethernet adapters can be shared. Other types of network adapters cannot be shared.
- ▶ IP forwarding is not supported on the Virtual I/O Server.
- ▶ The maximum number of virtual adapters can be any value from 2 to 65,536. However, if you set the maximum number of virtual adapters to a value higher than 1024, the logical partition might fail to activate, or the server firmware might require more system memory to manage the virtual adapters.

Sample scenario

The sample scenario in Figure 10-23 shows three client partitions (Partition 1 through Partition 3) running AIX, IBM i, and Linux as well as one Virtual I/O Server (VIOS). Each of the client partitions is defined with one virtual Ethernet adapter. The Virtual I/O Server has a Shared Ethernet Adapter (SEA) that bridges traffic to the external network.

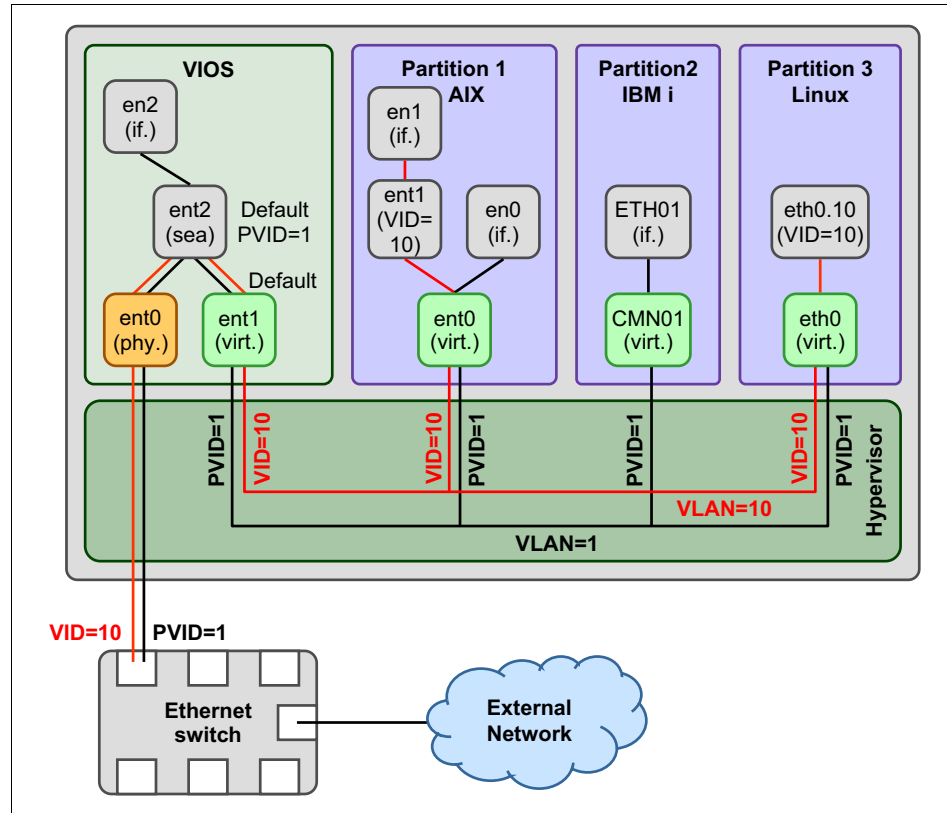


Figure 10-23 VLAN configuration example

Partition 2 is running IBM i, which does not support IEEE802.1Q VLAN tagging. Therefore it is using virtual Ethernet adapters with the Port Virtual LAN ID (PVID) only. This indicates the following conditions:

- ▶ The operating system running in such a partition is not aware of the VLANs.
- ▶ Only packets for the VLAN specified as PVID are received.
- ▶ Packets have their VLAN tag removed by the POWER Hypervisor before the partitions receive them.
- ▶ Packets sent by these partitions have a VLAN tag attached for the VLAN specified as PVID by the POWER Hypervisor.

In addition to the PVID, the virtual Ethernet adapters in Partition 1 and Partition 3 are also configured for VLAN 10. In the AIX partition a VLAN Ethernet adapter and network interface (en1) are configured through **smitty vlan**. On Linux, a VLAN Ethernet adapter eth0.10 is configured using the **vconfig** command.

From this, we can also make the following assumptions:

- ▶ Packets sent through network interfaces en1 and eth0.10 are tagged for VLAN 10 by the VLAN Ethernet adapter running in the client partition.
- ▶ Only packets for VLAN 10 are received by the network interfaces en1 and eth0.10.
- ▶ Packets sent through en0 or eth0 are not tagged by the operating system, but are automatically tagged for the VLAN specified as PVID by the POWER Hypervisor.
- ▶ Only packets for the VLAN specified as PVID are received by the network interfaces en0 and eth0.

In the configuration shown in Figure 10-23 on page 232, the Virtual I/O Server (VIOS) bridges both VLAN 1 and VLAN 10 through the Shared Ethernet Adapter (SEA) to the external Ethernet switch. But the Virtual I/O Server itself can only communicate with VLAN 1 through its network interface en2 attached to the SEA. Because this is associated with the PVID, VLAN tags are automatically added and removed by the POWER Hypervisor when sending and receiving packets to other internal partitions through interface en2.

Table 10-4 summarizes which partitions in the virtual Ethernet configuration from Figure 10-23 on page 232 can communicate with each other internally through which network interfaces.

Table 10-4 Inter-partition VLAN communication

Internal VLAN	Partition / network interface
1	Partition 1 / en0 Partition 2 / ETH0 Partition 3 / eth0 Virtual I/O Server / en2
10	Partition 1 / en1 Partition 3 / eth0.10

If the Virtual I/O Server is required to communicate with VLAN 10 as well, then it will need to have an additional Ethernet adapter and network interface with an IP address for VLAN 10, as shown on the left in Figure 10-24. A VLAN-unaware virtual Ethernet adapter with a PVID only, also shown on the left in Figure 10-24, will be sufficient; there is no need for a VLAN-aware Ethernet adapter (ent4), as shown in the center of Figure 10-24.

Only the simpler configuration with a PVID will be effective, because the Virtual I/O Server already has access to VLAN 1 through the network interface (en2) attached to the SEA (ent2). Alternatively, you can associate an additional VLAN Ethernet adapter (ent3) to the SEA (ent2), as shown on the right in Figure 10-24.

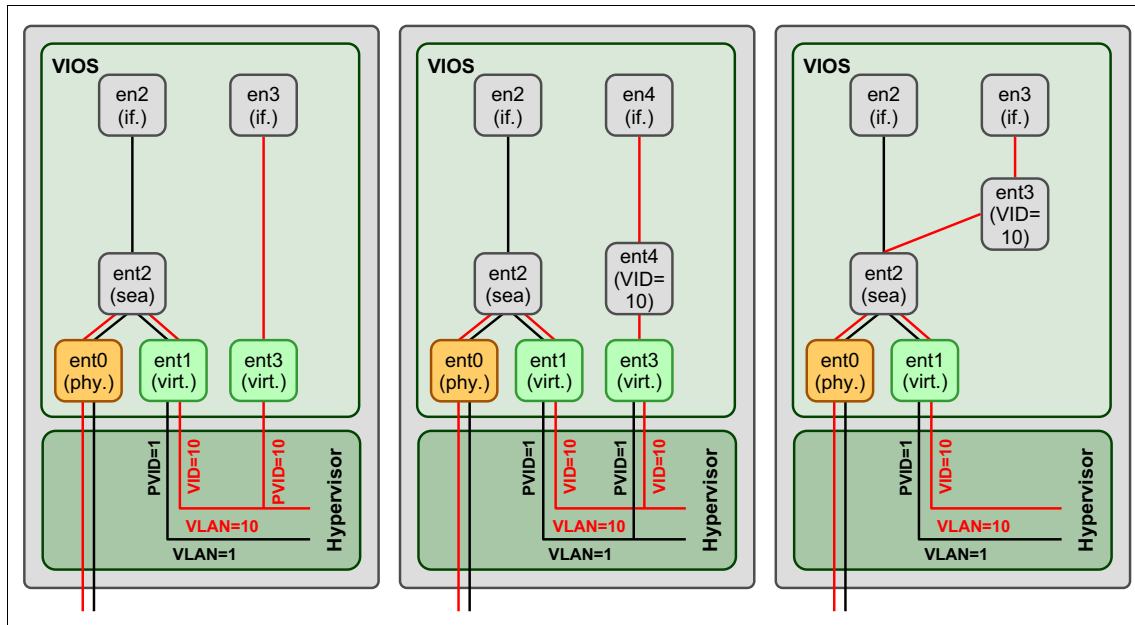


Figure 10-24 Adding virtual Ethernet adapters on the Virtual I/O Server for VLANs

Tip: Although it is possible to configure multiple IP addresses on a Virtual I/O Server, it can cause unexpected results because some commands of the command line interface make the assumption that there is only one.

An IP address is necessary on a Virtual I/O Server to allow communication with the HMC through RMC, which is a prerequisite to perform dynamic LPAR operations.

The Shared Ethernet Adapter (SEA) of Figure 10-23 on page 232 is configured with default PVID 1 and default adapter ent1. This means that untagged packets or packets with VID=1 that are received by the SEA from the external network are forwarded to adapter ent1. The virtual Ethernet adapter ent1 has the additional VID 10. Thus, packets tagged with VID 10 will be forwarded to ent1 as well.

The handling of outgoing traffic to the external network depends on the VLAN tag of the outgoing packets:

- ▶ Packets tagged with VLAN 1, which matches the PVID of the virtual Ethernet adapter ent1, are untagged by the POWER Hypervisor before they are received by ent1, bridged to ent0 by the SEA, and sent out to the external network.
- ▶ Packets tagged with a VLAN other than the PVID 1 of the virtual Ethernet adapter ent1, such as VID 10, are processed with the VLAN tag unmodified.

In the virtual Ethernet and VLAN configuration example of Figure 10-23 on page 232, the client partitions have access to the external Ethernet through the network interfaces en0, ETH0 and eth0 using PVID 1.

- ▶ Because packets with VLAN 1 are using the PVID, the POWER Hypervisor will remove the VLAN tags before these packets are received by the virtual Ethernet adapter of the client partition.
- ▶ Because VLAN 1 is also the PVID of ent1 of the SEA in the Virtual I/O Server, these packets will be processed by the SEA without VLAN tags and will be send out untagged to the external network.
- ▶ Therefore, VLAN-unaware destination devices on the external network will be able to receive the packets as well.

Partition 1 and Partition 3 have access to the external Ethernet through network interface en1 and eth0.10 to VLAN 10.

- ▶ These packets are sent out by the VLAN Ethernet adapters ent1 and eth0.10, tagged with VLAN 10, through the physical Ethernet adapter ent0.
- ▶ The virtual Ethernet adapter ent1 of the SEA in the Virtual I/O Server also uses VID 10 and will receive the packet from the POWER Hypervisor with the VLAN tag unmodified. The packet will then be sent out through ent0 with the VLAN tag unmodified.
- ▶ So, only VLAN-capable destination devices will be able to receive these.

Table 10-5 summarizes which partitions in the virtual Ethernet configuration from Figure 10-23 on page 232 can communicate with which external VLANs through which network interface.

Table 10-5 VLAN communication to external network

External VLAN	Partition / network interface
1	Partition 1 / en0 Partition 2 / ETH0 Partition 3 / eth0 Virtual I/O Server / en2
10	Partition 1 / en1 Partition 3 / eth0.10

If this configuration must be extended to enable Partition 4 to communicate with devices on the external network, but without making Partition 4 VLAN-aware, the following alternatives may be considered:

- ▶ An additional physical Ethernet adapter can be added to Partition 4.
- ▶ An additional virtual Ethernet adapter ent1 with PVID=1 can be added to Partition 4. Then Partition 4 will be able to communicate with devices on the external network using the default VLAN=1.
- ▶ An additional virtual Ethernet adapter ent1 with PVID=10 can be added to Partition 4. Then Partition 4 will be able to communicate with devices on the external network using VLAN=10.
- ▶ VLAN 2 can be added as additional VID to ent1 of the Virtual I/O Server partition, thus bridging VLAN 2 to the external Ethernet, just like VLAN 10. Then Partition 4 will be able to communicate with devices on the external network using VLAN=2. This will work only if VLAN 2 is also known to the external Ethernet and there are some devices on the external network in VLAN 2.
- ▶ Partition 3 can act as a router between VLAN 2 and VLAN 10 by enabling IP forwarding on Partition 3 and adding a default route by Partition 3 to Partition 4.

Throughput maximization

There are various ways to configure physical and virtual Ethernet adapters into Shared Ethernet Adapters to maximize throughput:

- ▶ Using Link Aggregation (EtherChannel), several physical network adapters can be aggregated. See 10.3.6, “Using Link Aggregation on the Virtual I/O Server” on page 248 for more details.
- ▶ Using several Shared Ethernet Adapters provides more queues and more performance.

10.3.5 Availability

PowerVM offers a range of configurations to keep the services availability. The following sections present some example scenarios.

Virtual Ethernet redundancy

In a single Virtual I/O Server configuration, communication to external networks ceases if the Virtual I/O Server loses connection to the external network. Client partitions will experience this disruption if they use the Shared Ethernet Adapter (SEA) as a means to access the external networks. Communication through the SEA is, for example, suspended when the physical network adapter in the Virtual I/O Server fails or loses connectivity to the external network due to a switch failure.

Another reason might be a planned shutdown of the Virtual I/O Server for maintenance purposes. Communication resumes as soon as the Virtual I/O Server regains connectivity to the external network. Internal communication between partitions through virtual Ethernet connections continues unaffected while access to the external network is unavailable. Virtual I/O clients do not have to be rebooted or otherwise reconfigured to resume communication through the SEA. The clients are similarly affected as when unplugging and replugging an uplink of a physical Ethernet switch.

If the temporary failure of communication with external networks is unacceptable, more than a single forwarding instance and some function for failover has to be implemented in the Virtual I/O Server.

Support: The Integrated Virtualization Manager (IVM) supports a single Virtual I/O Server. This section only applies to systems managed by the Hardware Management Console (HMC).

There are several approaches to achieve high availability for shared Ethernet access to external networks. Most commonly used are Shared Ethernet Adapter Failover and Network Interface Backup, which are described in detail in the following sections.

There are other approaches to achieve high availability for shared Ethernet access by taking advantage of configurations that are also used in physical network environments, such as these:

- ▶ IP Multipathing with Dead Gateway Detection (DGD) or Virtual IP Addresses (VIPA) and dynamic routing protocols, such as Open Shortest Path First (OSPF)
- ▶ IP Address Takeover (IPAT), with High Availability Cluster Management or Automation Software, such as PowerHA SystemMirror for AIX or Tivoli System Automation (TSA)

Shared Ethernet Adapter failover

Shared Ethernet Adapter failover offers Ethernet redundancy to the client at the Virtual I/O Server level. In a SEA failover configuration, two Virtual I/O Servers have the bridging functionality of the Shared Ethernet Adapter. They use a control channel to determine which of them is supplying the Ethernet service to the client. If one SEA loses access to the external network through its physical Ethernet adapter or one Virtual I/O Server is shut down for maintenance, it will automatically fail over to the other Virtual I/O Server SEA. You can also trigger a manual failover.

The client partition gets one virtual Ethernet adapter bridged by two Virtual I/O Servers. The client partition has no special protocol or software configured and uses the virtual Ethernet adapter as if it was bridged by only one Virtual I/O Server.

Shared Ethernet Adapter failover supports IEEE 802.1Q VLAN tagging just like the basic SEA feature.

As shown in Figure 10-25, both Virtual I/O Servers attach to the same virtual and physical Ethernet networks and VLANs, and both virtual Ethernet adapters of both SEAs will have the *access external network* (in a later HMC version, it is *Use this adapter for Ethernet bridging*) flag enabled and a *trunk priority* (in a later HMC version, it is *priority*) set.

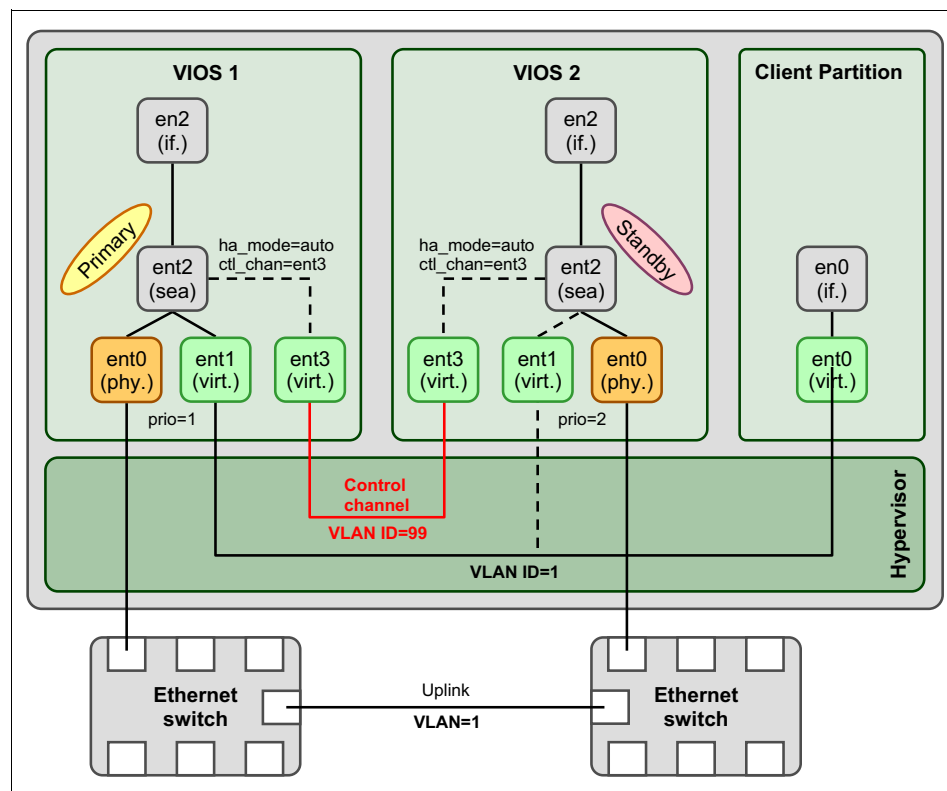


Figure 10-25 Basic SEA failover configuration

An additional virtual Ethernet connection has to be set up as a separate VLAN between the two Virtual I/O Servers and must be attached to the Shared Ethernet Adapter (SEA) as a *control channel*, not as a regular member of the SEA. This VLAN serves as a channel for the exchange of keep-alive or heartbeat messages between the two Virtual I/O Servers and therefore controls the failover of the bridging functionality. No network interfaces have to be attached to the control channel Ethernet adapters. The control channel adapter must be placed on a dedicated VLAN that is not used for anything else.

You must select different priorities for the two SEAs by setting all virtual Ethernet adapters of each SEA to that priority value. The priority value defines which of the two SEAs will be the primary (active) and which will be the backup (standby). The lower the *priority value*, the higher the priority, thus priority=1 means highest priority.

Support: SEA failover configurations are only supported on dual Virtual I/O Server configuration.

There may also be some types of network failures that will not trigger a failover of the SEA, because keep-alive messages are only sent over the control channel. No keep-alive messages are sent over other SEA networks, especially not over the external network. The SEA failover feature can be configured to periodically check the reachability of a given IP address. The SEA will periodically ping this IP address in order to detect some other network failures. This is similar to the IP address to ping that can be configured with Network Interface Backup.

Important: The Shared Ethernet Adapters must have network interfaces, with IP addresses associated, to be able to use this periodic reachability test. These IP addresses have to be unique and you have to use different IP addresses on the two SEAs.

There are basically four different cases that will initiate a SEA failover:

- ▶ The standby SEA detects that keep-alive messages from the active SEA are no longer received over the control channel.
- ▶ The active SEA detects that a loss of the physical link is reported by the physical Ethernet adapter's device driver.
- ▶ On the Virtual I/O Server with the active SEA, a manual failover can be initiated by setting the active SEA to standby mode.
- ▶ The active SEA detects that it cannot ping a given IP address anymore.

An end of the keep-alive messages will occur when the Virtual I/O Server with the primary SEA has been shut down or halted, has stopped responding, or has been deactivated from the HMC.

Important: You might experience up to a 30-second failover delay when using SEA failover. The behavior depends on the network switch and the spanning tree settings. Any of the following three hints can help in reducing this delay to a minimum:

- ▶ For all AIX client partitions, set up Dead Gateway Detection (DGD) on the default route:
 - a. Set up DGD on the default route:

```
# route change default -active_dgd
```
 - b. Add the command **route change default -active_dgd** to the `/etc/rc.tcpip` file to make this change permanent.
 - c. Set interval between pings of a gateway by DGD to 2 seconds (default is 5 seconds; setting this parameter to 1 or 2 seconds will allow faster recovery):

```
# no -p -o dgd_ping_time=2
```
- ▶ On the network switch, enable PortFast if Spanning Tree is on or disable Spanning Tree.
- ▶ On the network switch, set the channel group for your ports to **Active** if they are currently set to **Passive**.

Figure 10-26 shows an alternative setup where the IP address of the Virtual I/O Servers is configured on a separate physical Ethernet adapter.

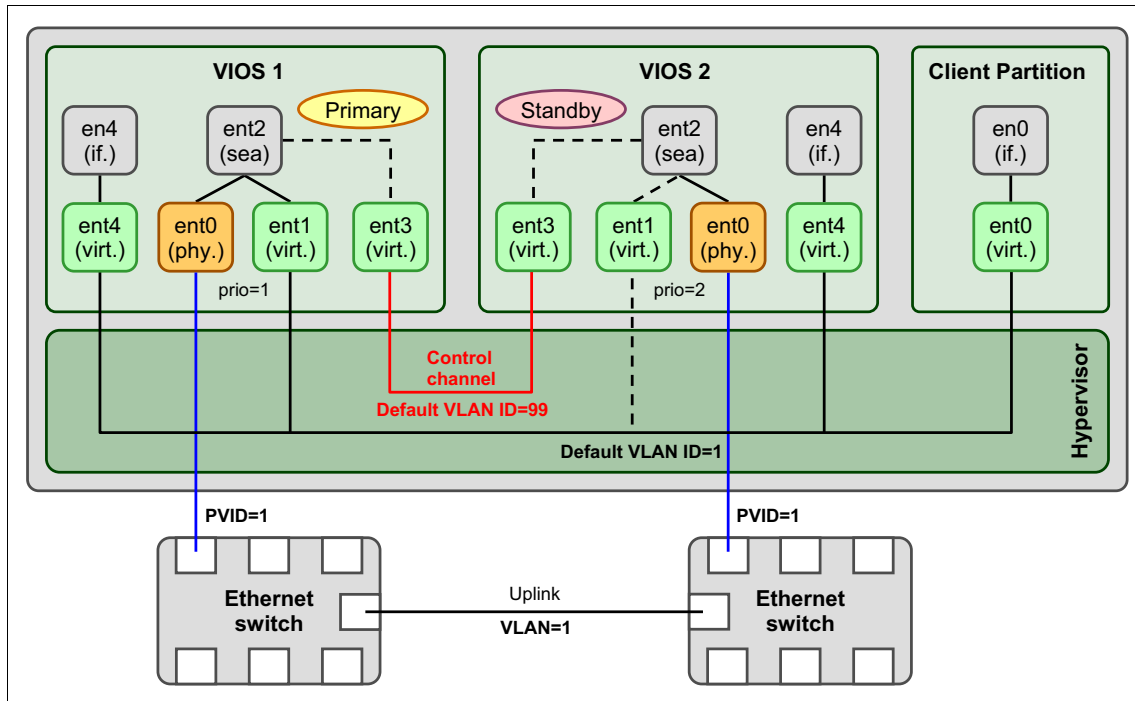


Figure 10-26 Alternative configuration for SEA failover

In this configuration, it is not possible for the SEAs to periodically ping a given IP address. It is best to associate the interface and IP address to the SEA, as shown in Figure 10-25 on page 238.

Network Interface Backup in the client partition

Network Interface Backup (NIB) in the client partition can be used to achieve network redundancy when using two Virtual I/O Servers. An EtherChannel with only one primary adapter and one backup adapter is said to be operating in Network Interface Backup mode.

Figure 10-27 shows an NIB setup for an AIX client partition. The client partition uses two virtual Ethernet adapters to create an EtherChannel that consists of one primary adapter and one backup adapter. The interface is defined on the EtherChannel. If the primary adapter becomes unavailable, the Network Interface Backup switches to the backup adapter.

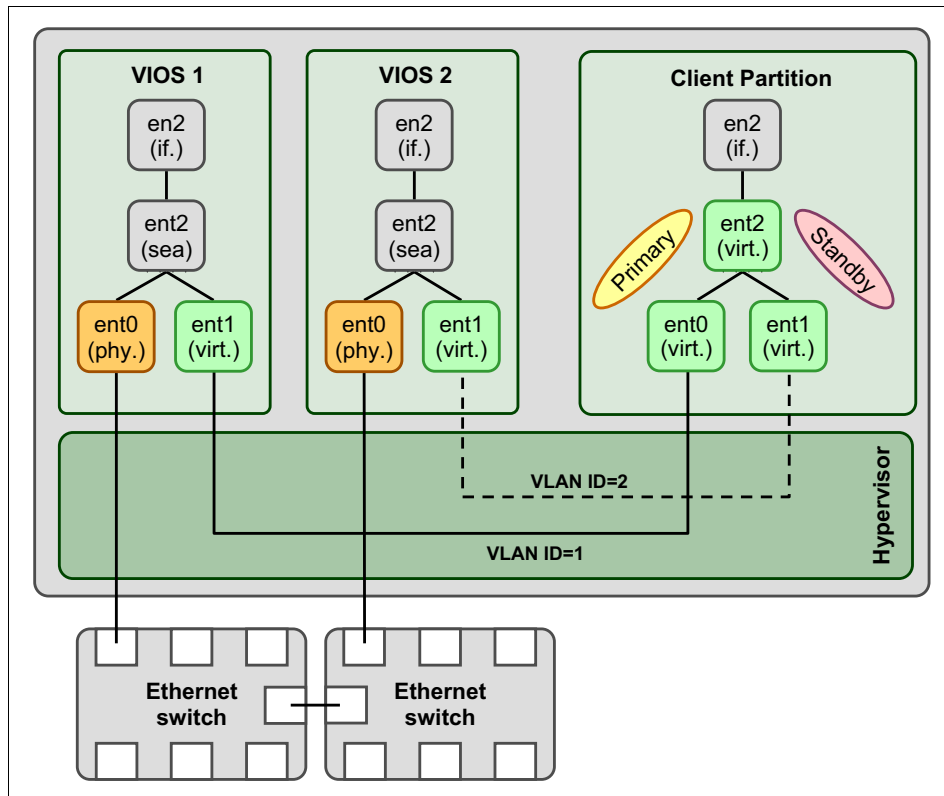


Figure 10-27 Network redundancy using two Virtual I/O Servers and NIB

A Link Aggregation of more than one active virtual Ethernet adapter is not supported. Only one primary virtual Ethernet adapter plus one backup virtual Ethernet adapter are supported. To increase the bandwidth of a virtual Ethernet adapter, Link Aggregation has to be done on the Virtual I/O Server as described in “Using Link Aggregation on the Virtual I/O Server” on page 248.

When configuring NIB in a client partition, each virtual Ethernet adapter has to be configured on a different VLAN. It is not possible to configure additional VLANs. The two different internal VLANs are then bridged to the same external VLAN, as shown in Figure 10-27 on page 242.

Important: When using NIB with virtual Ethernet adapters on AIX, it is mandatory to use the *ping-to-address* feature to be able to detect network failures, because there is no hardware link failure for virtual Ethernet adapters to trigger a failover to the other adapter.

For IBM i, an equivalent solution to NIB can be implemented using virtual IP address (VIPA) failover with a virtual-to-virtual Ethernet adapter failover script.

- ▶ For setup details on AIX, see 16.3.3, “EtherChannel Backup in the AIX client” on page 603.
- ▶ For setup details on IBM i, see 16.3.4, “IBM i virtual IP address failover for virtual Ethernet adapters” on page 608.
- ▶ For setup details on Linux, see 16.3.5, “Linux Ethernet connection bonding” on page 611.

SEA failover with load sharing

The Virtual I/O Server version 2.2.1.0 or later provides a load sharing function to enable to use the bandwidth of the backup Shared Ethernet Adapter (SEA).

The SEA failover configuration provides redundancy by configuring a primary and backup SEA pair on Virtual I/O Servers (VIOS). The backup SEA is in standby mode, and is used when the primary SEA fails. The bandwidth of the backup SEA is not used in normal operation.

Figure 10-28 shows a basic SEA failover configuration. All network packets of all Virtual I/O clients are bridged by the primary Virtual I/O Server.

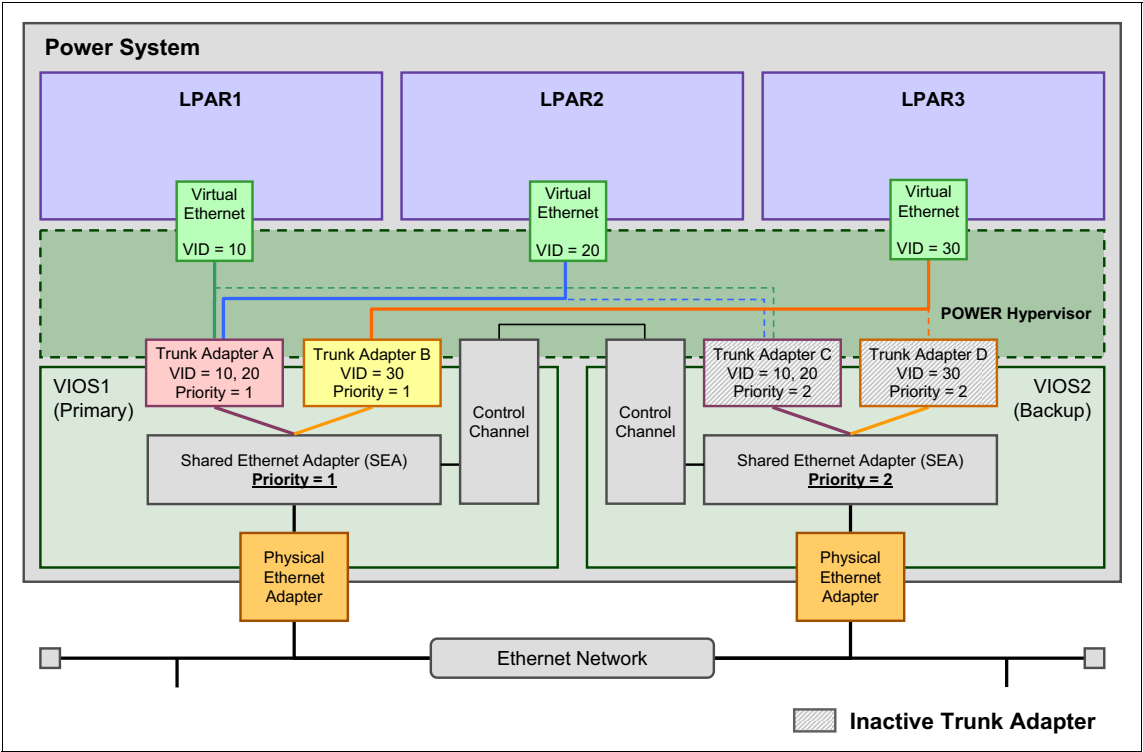


Figure 10-28 SEA failover Primary-Backup configuration

On the other hand, SEA failover with Load Sharing makes effective use of the backup SEA bandwidth, as shown in Figure 10-29. In this example, network packets of LPAR1 and LPAR2 are bridged by VIOS2, and LPAR3 is bridged by VIOS1.

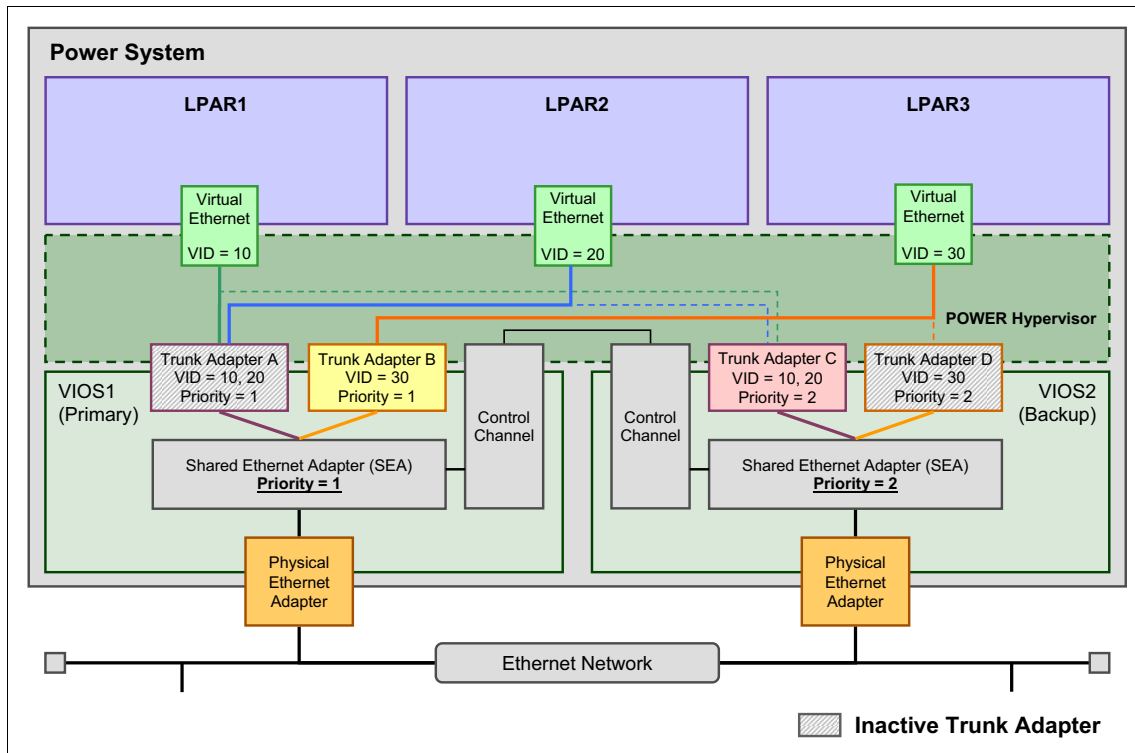


Figure 10-29 SEA failover with Load Sharing

Prerequisites and requirements for SEA failover with Load Sharing are as follows:

- ☐ Both primary and backup Virtual I/O Servers are at Version 2.2.1.0 or later.
- ☐ Two or more trunk adapters are configured for the primary and backup SEA pair.
- ☐ Load Sharing mode must be enabled on both primary and backup SEA pair.
- ☐ The VLAN definitions of the trunk adapters are identical between the primary and backup SEA pair.

Important: You need to set the same priority to all trunk adapters under one SEA. The primary and backup priority definitions are set at the SEA level, not at trunk adapters level.

These requirements are reflected in the sample SEA failover with Load Sharing configuration shown by Figure 10-29 on page 245:

- ▶ Both VIOS1 and VIOS2 should be at Version 2.2.1.0, or later.
- ▶ Two trunk adapters, Adapter A and B, are configured on the primary SEA on VIOS1, and Adapter C and D are configured on the backup SEA on VIOS2.
- ▶ All of the VLAN definitions of trunk adapters match. The primary SEA on VIOS1 has Adapter A with VLANs 10 and 20, and the backup SEA on VIOS2 has Adapter C with VLANs 10 and 20. Adapter B and Adapter D is the same.

Advantages of SEA failover

The SEA failover provides the following advantages over Network Interface Backup:

- ▶ SEA failover is implemented on the Virtual I/O Server, which simplifies the virtual network administration.
- ▶ The client partitions only require a single virtual Ethernet adapter and VLAN with no failover logic implemented, making the configuration of clients easier.
- ▶ When using the Network Interface Backup approach, the client partition configuration is more complex because all clients have to configure a second virtual Ethernet adapter on a different VLAN and a link aggregation adapter with the NIB feature.
- ▶ SEA failover has the added support of IEEE 802.1Q VLAN tagging.
- ▶ SEA failover simplifies NIM installation because only a single virtual Ethernet device is required on the client partition. The Ethernet configuration does not need to be modified after a NIM installation.
- ▶ Only the SEAs send out periodic ping requests for checking the network availability. With NIB every client partition will send out ping requests, resulting in more network traffic.

Use SEA failover rather than the Network Interface Backup option for the following situations:

- ▶ You want to provide virtual Ethernet redundancy with minimum setup and management effort.
- ▶ You use VLAN tagging.
- ▶ You are running IBM i or Linux operating system.
- ▶ You do not need load balancing per Shared Ethernet Adapter between the primary and standby Virtual I/O Servers.

Tip: In most cases, the advantages of SEA failover will outweigh those of NIB, so SEA failover can be the default approach to provide high-availability for bridged access to external networks.

Advantages of Network Interface Backup

Network Interface Backup (NIB) can provide better resource utilization as a result of the following features:

- ▶ With NIB, you can distribute the clients over both Shared Ethernet Adapters in such a way that half of them will use the Shared Ethernet Adapter on the first Virtual I/O Server as the primary adapter, and the other half will use the Shared Ethernet Adapter on the second Virtual I/O Server as the primary adapter. This enables the bandwidth of the physical Ethernet adapters in both the Virtual I/O Servers Shared Ethernet Adapters to be used concurrently by different client partitions. You are able to do this using additional pairs for additional VLANS and also requiring additional hardware.
- ▶ With SEA failover, only one of the two Shared Ethernet Adapters is actively used at any time, while the other Shared Ethernet Adapter is only a standby. Therefore, the bandwidth of the physical Ethernet adapters of the standby Shared Ethernet Adapter is not used.

Using NIB might be preferable over the Shared Ethernet Adapter failover option in the following situations:

- ▶ You want to load balance client partitions to use both Virtual I/O Servers.
- ▶ You are not using VLAN tagging.

Tip: SEA failover and NIB have a common behavior: they do not check the reachability of the specified IP address through the backup path as long as the primary path is active. That is because the virtual Ethernet adapter is always connected and there is no *link up* event as is the case with physical adapters. You do not know if you really have an operational backup until your primary path fails.

10.3.6 Using Link Aggregation on the Virtual I/O Server

Link Aggregation is a network port aggregation technology that allows several Ethernet adapters to be aggregated together to form a single pseudo-Ethernet adapter. This technology can be used on the Virtual I/O Server to increase the bandwidth compared to when using a single network adapter and avoid bottlenecks when sharing one network adapter among many client partitions.

The main benefit of a Link Aggregation is that it has the network bandwidth of all of its adapters in a single network presence. If an adapter fails, the packets are automatically sent to the next available adapter without disruption to existing user connections. The adapter is automatically returned to service on the Link Aggregation when it recovers. Thus, Link Aggregation also provides some degree of increased availability. A link or adapter failure will lead to a performance degradation, but not a disruption.

Depending on the manufacturer, Link Aggregation is not a complete high-availability networking solution because all the aggregated links must connect to the same switch. By using a backup adapter, you can add a single additional link to the Link Aggregation, which is connected to a different Ethernet switch with the same VLAN. This single link will only be used as a backup.

As an example for Link Aggregation, ent0 and ent1 can be aggregated to ent2. The system considers these aggregated adapters as one adapter. Interface en2 will then be configured with an IP address. Therefore, IP is configured as on any other Ethernet adapter. In addition, all adapters in the Link Aggregation are given the same hardware (MAC) address, so they are treated by remote systems as though they were one adapter.

Two variants of Link Aggregation are supported:

- ▶ Cisco EtherChannel (EC)
- ▶ IEEE 802.3ad Link Aggregation (LA)

While EC is a Cisco-specific implementation of adapter aggregation, LA follows the IEEE 802.3ad standard. Table 10-6 shows the main differences between EC and LA.

Table 10-6 Main differences between EC and LA aggregation

Cisco EtherChannel (EC)	IEEE 802.3ad Link Aggregation (LA)
Cisco-specific.	Open Standard.
Requires switch configuration.	Little, if any, configuration of switch required to form aggregation. Some initial setup of the switch might be required.
Supports different packet distribution modes.	Supports only standard distribution mode.

Using IEEE 802.3ad Link Aggregation allows for the use of Ethernet switches which support the IEEE 802.3ad standard but may not support EtherChannel. The benefit of EtherChannel is the support of different packet distribution modes. This means it is possible to influence the load balancing of the aggregated adapters. In the remainder of this publication, we use *Link Aggregation* where possible because that is considered a more universally understood term.

Note: When using IEEE 802.3ad Link Aggregation, ensure that your Ethernet switch hardware supports the IEEE 802.3ad standard. With VIOS version 2.2, configuring an Ethernet interface to use the 802.3ad mode requires that the Ethernet switch ports also be configured in IEEE 802.3ad mode.

Important: In previous versions of Virtual I/O Server, the implementation of the IEEE 802.3ad protocol did not require Ethernet switch ports to be configured to use 802.3ad protocol. Virtual I/O Server version 2.2 requires the corresponding Ethernet switch ports be configured in IEEE 802.3ad mode when the Ethernet interface is operating in the 802.3ad mode. When planning to upgrade or migrate to Virtual I/O Server 2.2, ensure that any Ethernet switch ports in use by an 802.3ad Link Aggregation are configured to support the 802.3ad protocol.

Figure 10-30 shows the aggregation of two plus one adapters to a single pseudo-Ethernet device, including a backup feature.

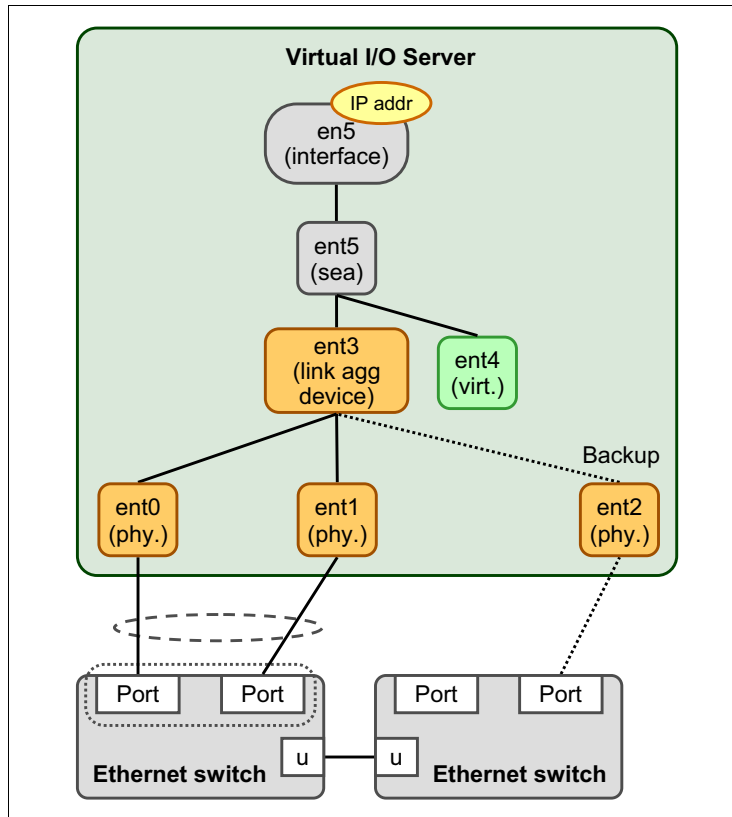


Figure 10-30 Link Aggregation (EtherChannel) on the Virtual I/O Server

The Ethernet adapters ent0 and ent1 are aggregated for bandwidth and must be connected to the same Ethernet switch, while ent2 connects to a different switch, but is only used for backup, for example, if the main Ethernet switch fails. The adapters ent0 and ent1 are now exclusively accessible through the pseudo-Ethernet adapter ent5 and its interface en5. You cannot, for example, attach a network interface en0 to ent0, as long as ent0 is a member of an EtherChannel or Link Aggregation.

Support: A Link Aggregation or EtherChannel of virtual Ethernet adapters is not supported. But you can use the Network Interface Backup feature of Link Aggregation with virtual Ethernet adapters.

A Link Aggregation with only one primary Ethernet adapter and one backup adapter is said to be operating in Network Interface Backup (NIB).

10.3.7 QoS

The Shared Ethernet Adapter is capable of enforcing Quality of Service (QoS), based on the IEEE 802.1q standard. This section explains how QoS works for SEA and how it can be configured.

SEA QoS provides a means whereby the VLAN tagged egress traffic is prioritized among 7 priority queues. However, note that QoS only comes into play when contention is present.

Each SEA instance has a certain number of threads (currently 7) for multiprocessing. Each thread will have 9 queues to take care of network jobs. Each queue will take care of jobs at a different priority level. One queue is kept aside that is used when QoS is disabled.

Important: QoS works only for *tagged* packets; that is, all packets emanating from the VLAN pseudo device of the virtual I/O client. Therefore, because virtual Ethernet does not tag packets, its network traffic cannot be prioritized. The packets will be placed in queue 0, which is the default queue at priority level 1.

Each thread will independently follow the same algorithm to determine from which queue to send a packet. A thread will sleep when there are no packets on any of the 9 queues.

Note the following points:

- ▶ If QoS is enabled, SEA will check the priority value of all tagged packets and put that packet in the corresponding queue.
- ▶ If QoS is *not* enabled, then regardless of whether the packet is tagged or untagged, SEA will ignore the priority value and place all packets in the disabled queue. This will ensure that the packets being enqueued while QoS is disabled will not be sent out of order when QoS is enabled.

When QoS is enabled, there are two algorithms to schedule jobs: strict mode and loose mode.

Strict mode

In strict mode, all packets from higher priority queues will be sent before any from a lower priority queue. The SEA will examine the highest priority queue for any packets to send out. If there are any packets to send, the SEA will send that packet. If there are no packets to send in a higher priority queue, the SEA will then check the next highest priority queue for any packets to send out and so on.

After sending out a packet from the highest priority queue with packets, the SEA will start the algorithm over again. This allows for high priorities to always be serviced before those of the lower priority queues.

Loose mode

It is possible, in strict mode, that lower priority packets will never be serviced if there are always higher priorities. To address this issue, the loose mode algorithm was devised.

With loose mode, if the number of bytes allowed has already been sent out from one priority queue, then the SEA will check all lower priorities at least once for packets to send before sending out packets from the higher priority again.

When initially sending out packets, SEA will check its highest priority queue. It will continue to send packets out from the highest priority queue until either the queue is empty or the cap is reached. After *either* of those two conditions has been met, SEA will then move on to service the next priority queue. It will continue using the same algorithm until either of the two conditions have been met in that queue. At that point, it would move on to the next priority queue. On a fully saturated network, this would allocate certain percentages of bandwidth to each priority. The caps for each priority are distinct and non-configurable.

A cap is placed on each priority level so that after a number of bytes is sent for each priority level, the following level is serviced. This method ensures that all packets are eventually sent. More important traffic is given less bandwidth with this mode than with strict mode. However, the caps in loose mode are such that more bytes are sent for the more important traffic, so it still gets more bandwidth than less important traffic. Set loose mode using this command:

```
chdev -dev -attr qos_mode=loose
```

The cap for each priority level is shown in Table 10-7.

Table 10-7 Cap values for loose mode

Priority	Cap in KB
1	256
2	128
0	64
3	32
4	16
5	8
6	4
7	2

10.3.8 Performance considerations

When using virtual networking, there are some performance implications to consider. Networking configurations are very site specific, therefore there are no guaranteed rules for performance tuning.

With Virtual and Shared Ethernet Adapter, we have the following considerations:

- ▶ The use of virtual Ethernet adapter in a partition does not increase its CPU requirement. High levels of network traffic within a partition will increase CPU utilization, however this behavior is not specific to virtual networking configurations.
- ▶ The use of Shared Ethernet Adapter in a Virtual I/O Server will increase the CPU utilization of the partition due to the bridging functionality of the SEA.
- ▶ Use the threading option on the SEA when the Virtual I/O Server is also hosting virtual SCSI.
- ▶ SEA configurations using 10 Gb/sec physical adapters can be demanding on CPU resources within the Virtual I/O Server. Consider using physical or dedicated shared CPUs on Virtual I/O Servers with these configurations.
- ▶ To reduce CPU processing overhead for TCP workloads on the Virtual I/O Server and client partitions and to better exploit the wire speed of 10 Gb Ethernet adapters enable large send offload on the client partition's interface (on the Virtual I/O Server it is enabled by default) and enable large receive offload on the SEA of the Virtual I/O Server.

Notes:

- ▶ For IBM i, large receive offload is supported with IBM i 7.1 TR5 and later.
 - ▶ Large receive offload by default is disabled on the Virtual I/O Server's *Shared Ethernet Adapter* as Linux typically cannot receive packages larger than their MTU.
-
- ▶ Consider the use of jumbo frames and increasing the MTU to 9000 bytes when using 10 Gb adapters if possible. Jumbo frames will enable higher throughput for less CPU cycles, however the external network also needs to be configured to support the larger frame size.

For further information about tuning network performance throughput, see *IBM PowerVM Virtualization Managing and Monitoring*, SG24-7590.



Server virtualization planning

This chapter discusses the points that you need to consider before implementing Dynamic Logical Partitioning (LPAR), Live Partition Mobility, or Partition Suspend/Resume.

It covers the following topics:

- ▶ Dynamic LPAR operations and dynamic resources planning
- ▶ Live Partition Mobility planning
- ▶ Suspend and Resume planning

11.1 Dynamic LPAR operations and dynamic resources planning

In both dedicated-processor partitions and micro-partitions, resources can be dynamically added and removed. In order to execute dynamic LPAR operations, each LPAR required communication access to the HMC or IVM.

11.1.1 Dedicated-processor partitions

Support for dynamic resources in dedicated-processor partitions provides for the dynamic movement of the following resources:

- ▶ One dedicated processor.
- ▶ A 16 MB memory region (dependent on the Logical Memory Block size).
- ▶ One I/O adapter slot (either physical or virtual).

It is only possible to dynamically add, move, or remove whole processors. When you dynamically remove a processor from a dedicated partition, the processor is then assigned to the shared processor pool.

11.1.2 Micro-partitions

The resources and attributes for micro-partitions include processor capacity, capped or uncapped mode, memory, and virtual or physical I/O adapter slots. All of these can be dynamically changed. For micro-partitions, it is possible to carry out the following operations dynamically:

- ▶ Remove, move, or add entitled capacity.
- ▶ Change the weight of an uncapped attribute.
- ▶ Add and remove virtual processors.
- ▶ Change mode between capped and uncapped.

11.1.3 Capacity on Demand

Several types of Capacity on Demand (CoD) are available to help meet changing resource requirements in an on-demand environment, by using resources that are installed on the system but that are not activated:

- ▶ Capacity Upgrade on Demand
- ▶ On/Off Capacity on Demand
- ▶ Utility Capacity on Demand
- ▶ Trial Capacity On Demand
- ▶ Capacity Backup
- ▶ Capacity backup for IBM i
- ▶ MaxCore/TurboCore and Capacity on Demand

Processor resource that are added using Capacity on Demand features are initially added to the default shared processor pool.

Memory resources that are added using Capacity on Demand features are added to available memory on the server. From there, they can be added to a Shared Memory Pool or as dedicated memory to a partition.

The IBM publication, *IBM Power 795 (9119-FHB) Technical Overview and Introduction*, REDP-4640, contains a concise summary of these features.

11.1.4 Planning for dynamic LPAR operations

Because the AIX and Linux operating systems of the partitions you want to perform dynamic LPAR operations need to communicate with HMC through the Resource Monitoring and Control (RMC) connection, you need to well plan your network for both partitions and the HMC.

Figure 11-1 shows the network connection between HMC, management system and the partitions. HMC needs to connect to both the hardware management private network and the customer network. The hardware management private network is for the communication between HMC and the POWER Hypervisor. The customer network is for the communication between HMC and the partitions. For security reason, you may consider to add firewall control for the customer network connection. In this situation, you should make sure the firewall is configured to allow RMC communication which uses TCP/UDP port number 657.

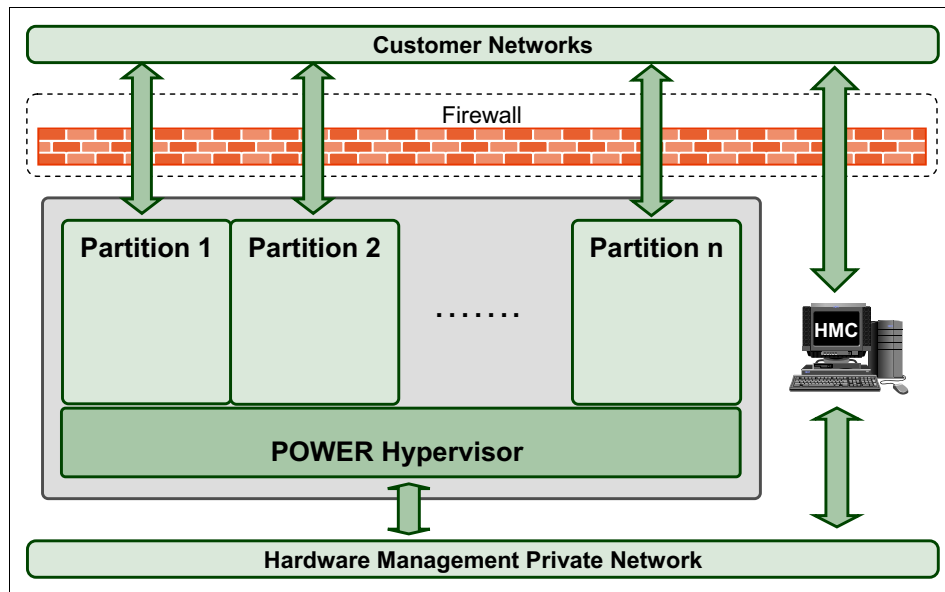


Figure 11-1 Network connections sample for Dynamic Logical Partitioning

Also, make sure that the RMC daemons are alive and working properly in your partitions before adjusting the resources dynamically. For more information, you can refer to the documents of the corresponding operating system.

To utilize the dynamic logical partitioning feature, you should also consider the following points when creating your partition:

- ▶ Select proper minimum and maximum value for memory and processor settings. You can only adjust them between the minimum and maximum value dynamically. If the value you want to set is beyond this range, you have to stop the partition before adjust it.
- ▶ For the physical adapters you plan to adjust dynamically later, you should select them as **Desired** resources when you creating the partition profile.
- ▶ If you want to add more virtual adapters dynamically later, you should set a proper value of **Maximum virtual adapters**, as shown in Figure 11-2. This value cannot be adjusted dynamically and you cannot add more virtual adapter if you reach this limitation.

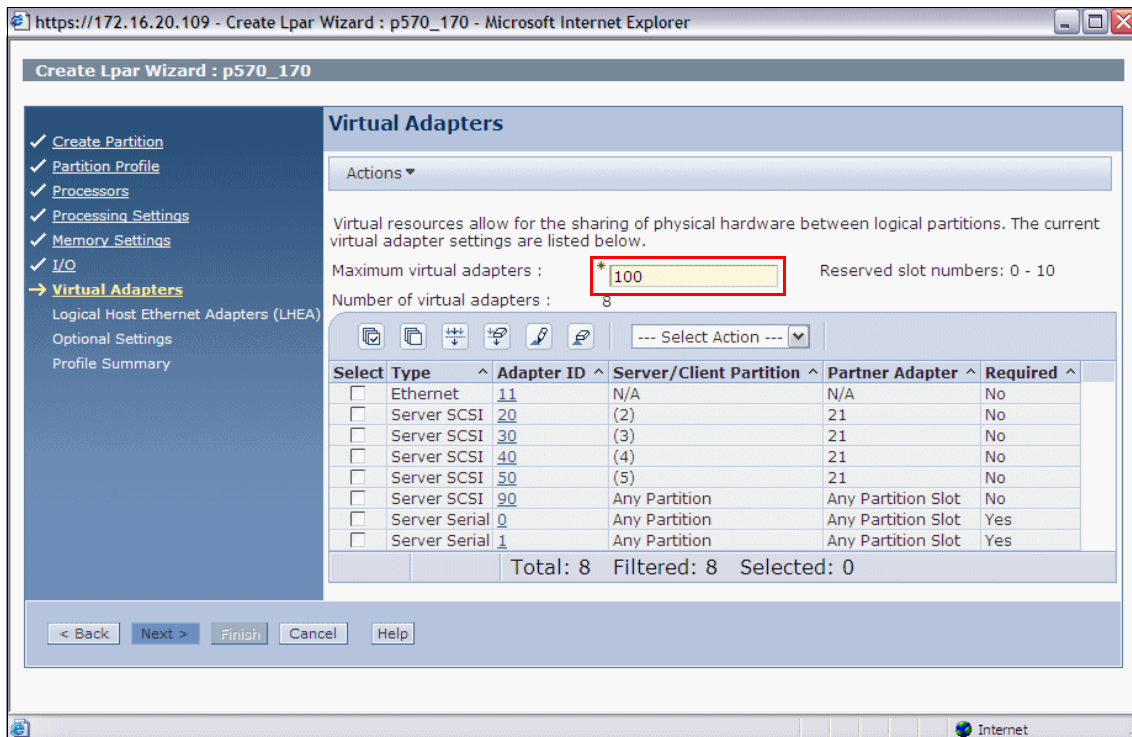


Figure 11-2 Set a proper Maximum virtual adapters value

11.1.5 Performing dynamic LPAR operations

There are several things you need to be aware of when carrying out dynamic LPAR operations that pertain to both virtualized and non-virtualized server resources:

- ▶ Make sure resources such as physical and virtual adapters being added and moved between partitions are not being used by other partitions. This means doing the appropriate cleanup on the client side by deleting them from the operating system or taking them offline by executing PCI hot-plug procedures if they are physical adapters.
- ▶ Memory or processors can be added to a running partition up to the maximum setting defined in the profile.
- ▶ For AIX and Linux the HMC must be able to communicate to the logical partitions over a network for RMC connections. IBM i doesn't use an RMC connection for dynamic LPAR but directly communicates with the POWER Hypervisor.
- ▶ Be aware of performance implications when removing memory or processors from logical partitions.
- ▶ Running applications can be dynamic LPAR-aware when doing dynamic resource allocation and deallocation so the system is able to resize itself and accommodate changes in hardware resources.
- ▶ For Linux additional filesets have to be installed to enable dynamic LPAR operations

For further information about performing and monitoring dynamic LPAR operations for AIX, IBM i, and Linux on IBM Power System servers, see SG24-7590.

11.2 Live Partition Mobility planning

Partition mobility provides the ability to move a logical partition from one system to another. Two types of partition migration can be distinguished:

- ▶ *Inactive partition migration* allows you to move a powered-off logical partition, including its partition profile, operating system and applications, from one system to another.
- ▶ *Active partition migration* is the ability to move a running logical partition, including its partition profile, operating system and applications, from one system to another without disrupting the operation of that logical partition.

- ▶ *Suspended partition migration* allows you to move a suspended logical partition, including its partition profile, operating system and applications from one system to another

When you have ensured that all requirements further described in the following sections are satisfied and all preparation tasks are completed, the HMC verifies and validates the Live Partition Mobility environment. If this validation turns out to be successful, then you can initiate the partition migration by using the wizard on the HMC graphical user interface (GUI) or through the HMC command-line interface (CLI).

PowerVM allows at least four concurrent migrations allowing up to eight concurrent migrations per Virtual I/O Server and up to 16 per system. For eight concurrent migrations on a Virtual I/O Server IBM recommends that you use a 10 Gbps network.

For detailed information about concurrent migration support and firmware compatibility between source and destination systems refer to the IBM Power Systems Hardware Information Center at:

<http://pic.dhe.ibm.com/infocenter/powersys/v3r1m5/index.jsp?topic=/p7hc3/p7hc3firmwaresupportmatrix.htm>

11.2.1 General requirements

These are the minimum requirements for the migration of a logical partition:

- ▶ Live Partition Mobility (LPM) is supported with the PowerVM Enterprise Edition for partitions running AIX 6.1, AIX 7.1, Red Hat Enterprise Linux version 5 Update 8 or later, or SuSE Linux Enterprise Server 10 Service Pack 4 or later, and IBM i 7.1 TR4. All operating requirements are detailed in 7.1.3, “Operating system requirements” on page 98.
- ▶ LPM for AIX and Linux requires two POWER6 and later technology-based systems running PowerVM Enterprise Edition with Virtual I/O Server (version 1.5.1.1 or higher) controlled by at least one HMC, one FSM, or each of them running the IVM.
- ▶ LPM for IBM i requires POWER7 and later based systems (firmware Ax730_087, Ax740_088, Flex System POWER7 compute node firmware AF763_042) running PowerVM Enterprise Edition with Virtual I/O Server (version 2.2.1.4-FP25-SP2) controlled by at least one HMC (V7R7.5), or FSM (1.2); IVM does not support LPM for IBM i.
- ▶ The destination system must have enough processor and memory resources to host the mobile partition (the partition profile that is running, as alternate production profiles might exist).

- ▶ The operating system, applications, and data of the mobile partition must reside on virtual storage on an external storage subsystem accessible by the Virtual I/O Servers on *both* source and destination system.
- ▶ No physical adapters may be used by the mobile partition during the migration.
- ▶ Migration of partitions using multiple Virtual I/O Servers is supported.
- ▶ The mobile partition's network and disk access must be virtualized by using one or more Virtual I/O Servers:
 - The Virtual I/O Servers on both systems must have a Shared Ethernet Adapter configured to bridge to the same Ethernet network used by the mobile partition.
 - The Virtual I/O Servers on both systems must be capable of providing virtual access to all disk resources the mobile partition is using.
 - The disks used by the mobile partition must be accessed through virtual SCSI, virtual Fibre Channel-based mapping, or both.
 - For AMS partitions and suspended partition migration, an equivalently sized paging space must be in the reserved storage pool on the destination server.

11.2.2 Migration capability and compatibility

The first step of any mobility operation is to validate the capability and compatibility of the source and destination systems. The high-level prerequisites for Live Partition Mobility are in the following list. If any of these elements are missing, a migration cannot occur:

- ▶ A ready source system that is capable of migration
- ▶ A ready destination system that is capable of migration
- ▶ Compatibility between the source and destination systems
- ▶ The source and destination systems, which might be under the control of a single HMC and might also include a redundant HMC. Note that both HMCs should be at the same level.
- ▶ A migratable, ready partition to be moved from the source system to the destination system. For an inactive migration, the partition must be powered down. However the partition must have been activated on the source system once and it must be capable of booting on the destination system.
- ▶ For active and suspended migrations, a Virtual I/O Server designated as a mover service partition on the source and destination systems

- ▶ An RMC connection to the Virtual I/O Server partitions as well as the migrating partition.

Before initiating the migration of a partition, the HMC verifies the capability and compatibility of the source and destination servers, and the characteristics of the mobile partition to determine whether or not a migration is possible.

The hardware, firmware, Virtual I/O Servers, mover service partitions, operating system, and HMC versions that are required for Live Partition Mobility along with the system compatibility requirements are described in on the course.

11.2.3 Readiness

Migration *readiness* is a dynamic partition property that changes over time.

A server that is running on battery power is not ready to receive a mobile partition; it cannot be selected as a destination for partition migration. A server that is running on battery power may be the source of a mobile partition; indeed, the fact that it is running on battery power may be the impetus for starting the migration.

11.2.4 Migratability

The term *migratability* refers to a partition's ability to be migrated and is distinct from partition readiness. A partition may be migratable but not ready. A partition that is not migratable may be made migratable with a configuration change. For active migration, consider whether a shutdown and reboot is required. When considering a migration, also consider the following additional prerequisites:

- ▶ General prerequisites:
 - The memory and processor resources required to meet the mobile partition's current entitlements must be available on the destination server.
 - The Virtual I/O Server must have enough virtual slots on the destination system.
 - The partition must not have any required dedicated physical adapters.
 - The partition must not have any logical host Ethernet adapters.
 - The partition is not a Virtual I/O Server.
 - The partition is not designated as a redundant error path reporting partition.
 - The partition does not have any of its virtual SCSI disks defined as logical volumes or files in any Virtual I/O Server. All virtual SCSI disks must be mapped to LUNs visible on a SAN.

Note: Virtual I/O Server Shared Storage Pools are supported for LPM but not for Partition Suspend and Resume.

- The partition has virtual Fibre Channel disks configured as described in 16.2.2, “Virtual Fibre Channel” on page 474.
- The partition is not part of an LPAR workload group. A partition can be dynamically removed from a group.
- The partition is not using huge pages.
- The partition is not using Barrier Synchronization Register (BSR) arrays.
- The partition has a unique name. A partition cannot be migrated if any partition exists with the same name on the destination server.
- An IBM i logical partition cannot be moved while an NPIV (virtual Fibre Channel) attached tape device is varied on.
- An IBM i partition cannot be hosting or hosted by IBM i. This includes the following restrictions:
 - The IBM i partition cannot be activated with a partition profile that has a virtual SCSI server adapter.
 - The IBM i partition cannot be activated with a partition profile that has a virtual SCSI client adapter that is hosted by another IBM i partition.
 - No virtual SCSI server adapters can be dynamically added to the IBM i partition.
 - No virtual SCSI client adapters that are hosted by another IBM i logical partition can be dynamically added to the logical partition being moved.
- The IBM i partition must not be assigned a virtual SCSI optical or tape device at the time of the operation.
- The property “restricted IO partition” must be set on the IBM i partition (this property can only be changed on a deactivated IBM i partition).
- In an *inactive migration only*, the following characteristics apply:
 - It is a partition in the Not Activated state.
 - It may use huge pages.
 - It may use the barrier synchronization registers.
 - Any physical I/O adapters configured as desired resources in an AIX or Linux mobile partition are automatically removed from the profile during migration.

Note: Consider recording the existing configuration of physical I/O resources of an AIX or Linux partition before inactive migration, to be able to easily add those resources to the partition again after migration.

- ▶ In a suspended migration only, the following characteristics apply:
 - It is a suspended partition
 - An equivalently sized paging space is in the reserved storage pool on the destination side

11.2.5 Infrastructure

Live Partition Mobility requires a specific infrastructure. Several platform components are involved, and in this section, they will be presented, highlighting the important points that should be assured to get a migration.

A migration operation requires a SAN and a LAN to be configured with their corresponding virtual SCSI, virtual Fibre Channel, VLAN, and virtual Ethernet devices. At least one Virtual I/O Server on both the source and destination systems must be configured as a mover service partitions for active or suspended migrations. The HMC must have RMC connections to the Virtual I/O Servers and a connection to the service processors on the source and destination servers. For an active migration, the HMC also needs RMC connections to the mobile partition and the mover service partitions. For a suspended partition migration an RMC connection is needed to the mover service partition.

Network

The Virtual network connectivity is crucial to get success to your migration. Some tasks must to be completed to ensure that your network configuration meets the minimal configuration for Live Partition Mobility.

PowerVM Live Partition Mobility can include one or more HMCs:

- ▶ The source system is managed by one HMC and the destination system is managed by a separate HMC. In this case, both the source HMC and the destination HMC must meet the following requirements:
 - The source HMC and the destination HMC must be connected to the same network so that they can communicate with each other.
 - An optional redundant HMC configuration is supported.

With regards to the partitions network configuration, you first have to create a shared Ethernet adapter on the Virtual I/O Server so that the client logical

partitions can access the external network without requiring a physical Ethernet adapter. Shared Ethernet adapters are required on both source and destination Virtual I/O Servers for all the external networks used by mobile partitions.

Note: Link Aggregation or EtherChannel can also be used as backing for the Shared Ethernet Adapter.

Perform the following steps on the source and destination Virtual I/O Servers:

1. Ensure that you connect the source and destination Virtual I/O Servers and the shared Ethernet adapter to the network.
2. Configure virtual Ethernet adapters for the source and destination Virtual I/O Server partitions. If virtual switches are available, be sure that the virtual Ethernet adapters on the source Virtual I/O Server is configured on a virtual switch that has the same name of the virtual switch that is used on the destination Virtual I/O Server.
3. Ensure that the mobile partition has a virtual Ethernet adapter.

Both the source and the target system must have an appropriate Shared Ethernet Adapter environment to host a moving partition. All virtual networks in use by the mobile partition on the source system must be available as virtual networks on the destination system.

VLANs defined by port virtual IDs (PVIDs) on the Virtual I/O Server have no meaning outside of an individual server as all packets are bridged untagged. It is possible for VLAN 1 on CEC 1 to be part of the 192.168.1.x network while VLAN 1 on CEC 2 is part of the 10.1.1.x network.

Because two networks are possible, you cannot verify whether VLAN 1 exists on both servers. You have to check whether VLAN 1 maps to the same network on both servers.

Figure 11-3 shows a basic hardware infrastructure enabled for Live Partition Mobility and that is using a single HMC. Each system is configured with a single Virtual I/O Server partition. The mobile partition has only virtual access to network and disk resources. The Virtual I/O Server on the destination system is connected to the same network and is configured to access the same disk space used by the mobile partition. For illustration purposes, the device numbers are all shown as zero, but in practice, they can vary considerably.

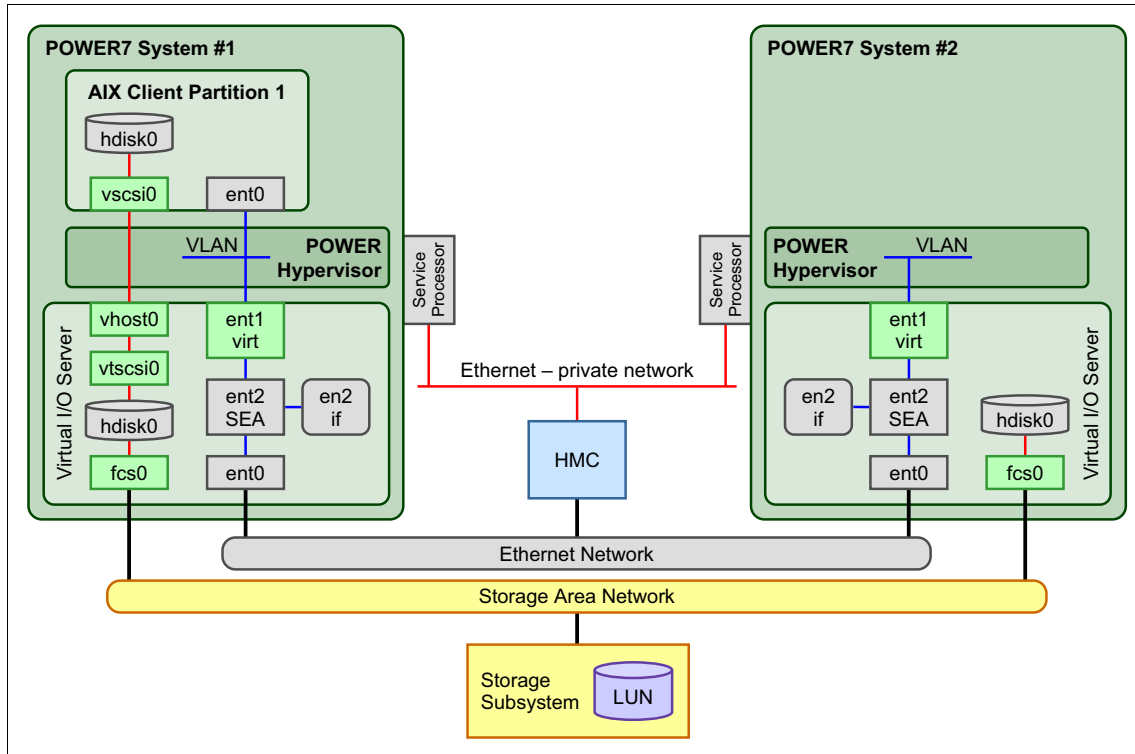


Figure 11-3 Hardware infrastructure enabled for Live Partition Mobility

The migration process creates a new logical partition on the destination system. This new partition uses the destination's Virtual I/O Server to access the same mobile partition's network and disks. During active migration, the state of the mobile partition is copied, as shown in Figure 11-4.

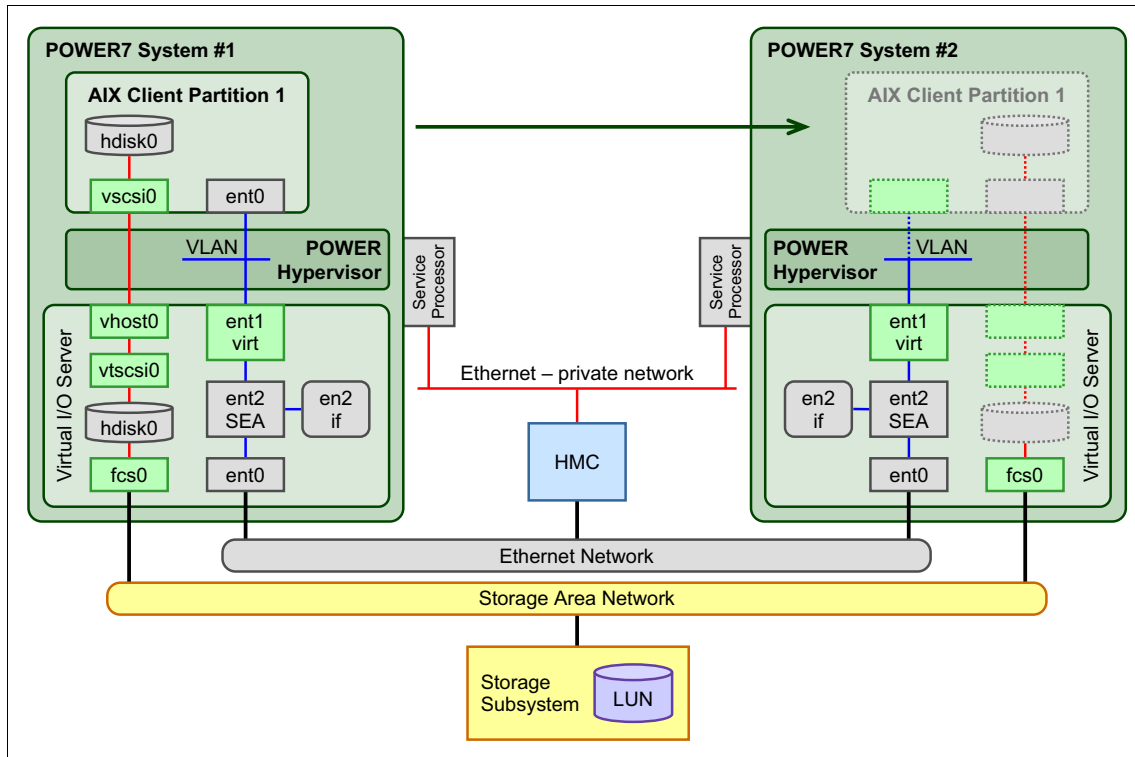


Figure 11-4 A mobile partition during migration

When the migration is complete, the source Virtual I/O Server is no longer configured to provide access to the external disk data. The destination Virtual I/O Server is set up to allow the mobile partition to use the storage. The final configuration is shown in Figure 11-5.

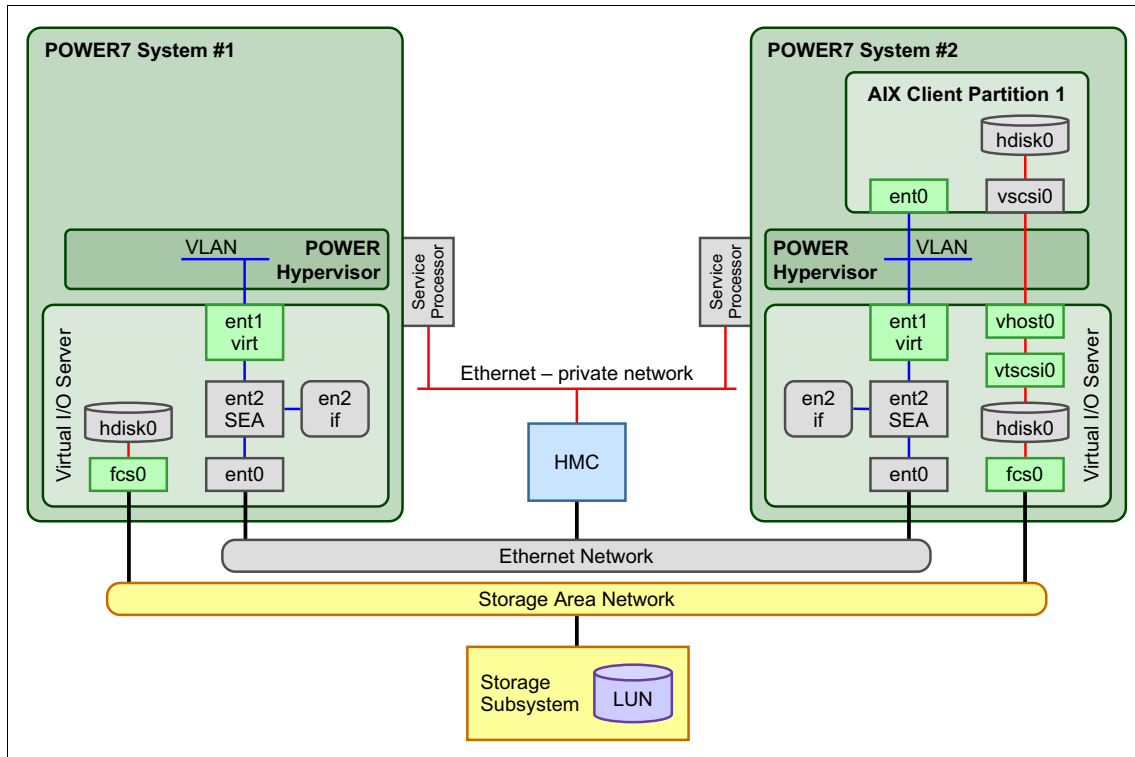


Figure 11-5 The final configuration after a migration is complete

The migrating partition uses the virtual LAN (VLAN) for network access. The VLAN must be bridged (if there is more than one, then it also has to be bridged) to a physical network using a shared Ethernet adapter in the Virtual I/O Server partition. Your LAN must be configured so that migrating partitions can continue to communicate with other necessary clients and servers after a migration is completed.

Storage

From a disk configuration perspective, it is required that one or more storage area networks (SAN) provide connectivity to all of the mobile partition's disks to the Virtual I/O Server partitions on *both* the source and destination servers.

The mobile partition accesses all migratable disks through virtual Fibre Channel, or virtual SCSI, or a combination of these devices. The LUNs used for virtual SCSI must be zoned and masked to the Virtual I/O Servers on *both* systems.

Virtual Fibre Channel LUNs should be configured as described in section 16.2.2, “Virtual Fibre Channel” on page 474.

11.2.6 Virtual I/O Server

This chapter collects useful Virtual I/O Server configurations showing the client and Virtual I/O Server behavior before and after the migration. The conception presented below offers you alternatives to select the better approach according what you need.

Several tasks must be completed to prepare the source and destination Virtual I/O Servers for Live Partition Mobility. At least one Virtual I/O Server logical partition must be installed and activated on both the source and destination systems. For Virtual I/O Server installation instructions, see Chapter 12, “I/O virtualization implementation” on page 311.

Virtual I/O Server version

To get all the advantages of the new Virtual I/O Servers features, use the version 2.2.2.0 or later on the source and destination Virtual I/O Servers.

This can be checked on the Virtual I/O Server by running the **ioslevel** command, as shown in Example 11-1.

Example 11-1 Output of the ioslevel command

```
$ ioslevel
2.2.2.0
```

If the source and destination Virtual I/O Servers do not meet the requirements, perform an upgrade:

- ▶ Dual Virtual I/O Server requirements:
 - At least one Virtual I/O Server at release level 1.5.1.1 or higher has to be installed both on the source and destination systems.
 - Ensure that at least one of the mover service partitions (MSP) is enabled on a source and destination Virtual I/O Server partition. The mover service partition is a Virtual I/O Server logical partition that is allowed to use its VASI adapter for communicating with the POWER Hypervisor.

There must be at least one mover service partition on both the source and destination Virtual I/O Servers for the mobile partition to participate in

active partition migration. If the mover service partition is disabled on either the source or destination Virtual I/O Server, the mobile partition can be migrated inactively.

To enable the source and destination mover service partitions using the HMC, you must have super administrator (such as hmcsuperadmin, as in the hscroot login) authority and complete the following steps:

1. In the navigation area, open **Systems Management** and select **Servers**.
2. In the contents area, open the source system.
3. Select the source Virtual I/O Server logical partition and select **Properties** on the task area.
4. On the **General** tab, select **Mover Service Partition**, and click **OK**.
5. Repeat these steps for the destination system.

Figure 11-6 shows the result of these actions.

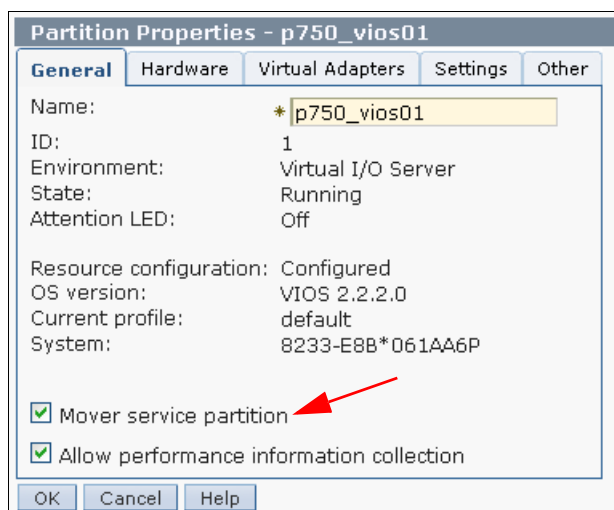


Figure 11-6 Enabling the Mover service partition MSP

- In addition to having the mover partition attribute set to TRUE, the source and destination mover service partitions communicate with each other over the network.

To determine the current release of the Virtual I/O Server and to see if an upgrade is necessary, use the `ioslevel` command.

More technical information about the Virtual I/O Server and latest downloads is available on the Virtual I/O Server website:

<http://www14.software.ibm.com/webapp/set2/sas/f/vios/download/home.html>

Dual Virtual I/O Servers

Multiple Virtual I/O Servers are often deployed in systems where there is a requirement for logical partitions to continue to use their virtual resources even during the maintenance of a Virtual I/O Server.

This discussion relates to the common practice of using more than one Virtual I/O Server to allow for concurrent maintenance, and is not limited to only two servers. Also, Virtual I/O Servers may be created to offload the mover services to a dedicated partition.

Live Partition Mobility does not make any changes to the network setup on the source and destination systems. It only checks that all virtual networks used by the mobile partition have a corresponding shared Ethernet adapter on the destination system. Shared Ethernet failover might or might not be configured on either the source or the destination systems.

Important: If you are planning to use shared Ethernet adapter failover, remember not to assign the Virtual I/O Server's IP address on the shared Ethernet adapter. Create another virtual Ethernet adapter and assign the IP address on it. Partition migration requires network connectivity through the RMC protocol to the Virtual I/O Server. The backup shared Ethernet adapter is always offline, and its associated IP address, if any.

When multiple Virtual I/O Servers are involved, multiple virtual SCSI and virtual Fibre Channel combinations are possible. Access to the same storage area network (SAN) disk may be provided on the destination system by multiple Virtual I/O Servers for use with virtual SCSI mapping. Similarly, multiple Virtual I/O Servers can provide access with multiple paths to a specific set of assigned LUNs for virtual Fibre Channel usage. Live Partition Mobility automatically manages the virtual SCSI and virtual Fibre Channel configuration if an administrator does not provide specific mappings.

The partition that is moving must keep the same number of virtual SCSI and virtual Fibre Channel adapters after migration and each virtual disk must remain connected to the same adapter or adapter set. An adapter's slot number can change after migration, but the same device name is kept by the operating system for both adapters and disks.

A migration can fail validation checks and is not started if the moving partition adapter and disk configuration cannot be preserved on the destination system. In this case, you are required to modify the partition configuration before starting the migration.

Tip: The best practice is to always perform a validation before performing a migration. The validation checks the configuration of the involved Virtual I/O Servers and shows you the configuration that will be applied. Use the validation menu on the HMC GUI or the `migr1par -o v` command.

Dual Virtual I/O Server and client mirroring

Dual Virtual I/O Server and client mirroring may be used when you have two independent storage subsystems providing disk space with data mirrored across them. It is not required that your mirroring use two independent storage subsystems, but it is recommended. With this setup, the partition can continue to run if one of the subsystems is taken offline.

If the destination system has two Virtual I/O Servers, one of them should be configured to access the disk space provided by the first storage subsystem; the other must access the second subsystem, as shown in Figure 11-7.

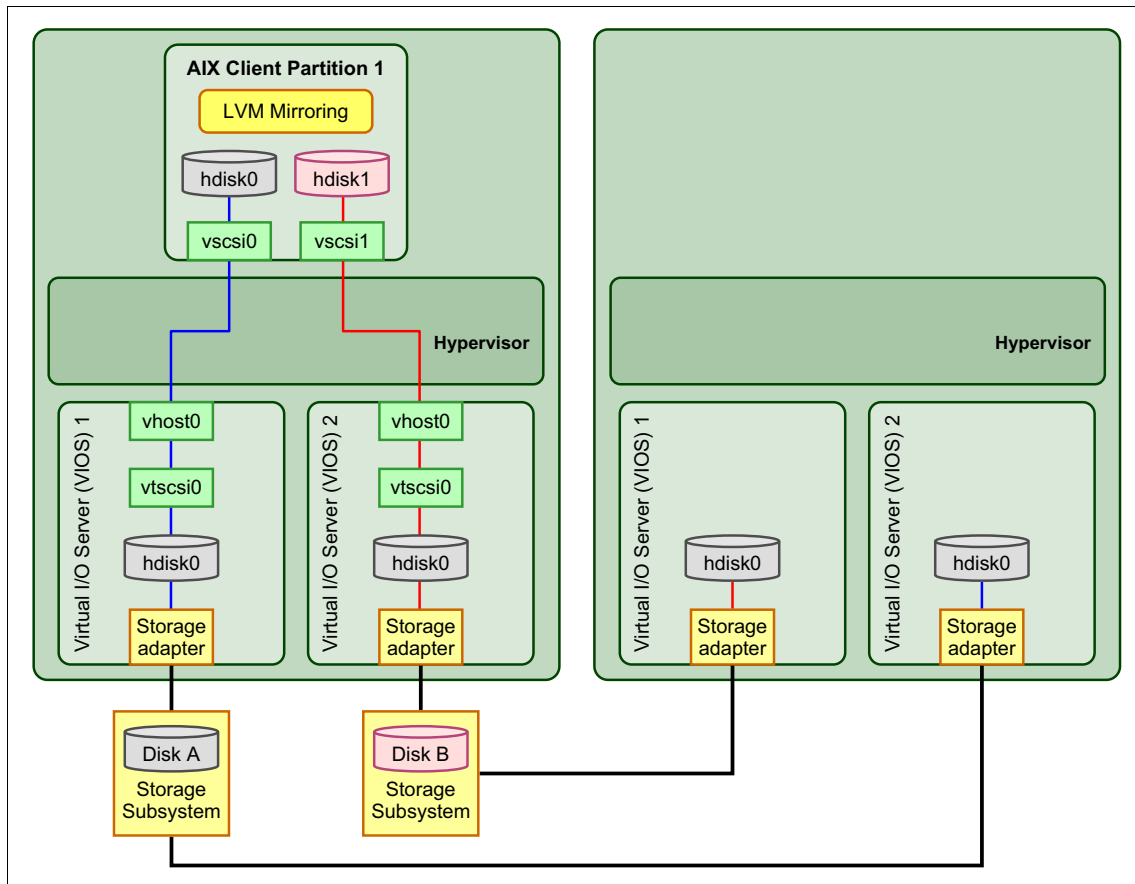


Figure 11-7 Dual VIOS and client mirroring to dual VIOS before migration

The migration process automatically detects which Virtual I/O Server has access to which storage and configures the virtual devices to keep the same disk access topology.

When migration is complete, the logical partition has the same disk configuration it had on previous system, still using two Virtual I/O Servers, as shown in Figure 11-8.

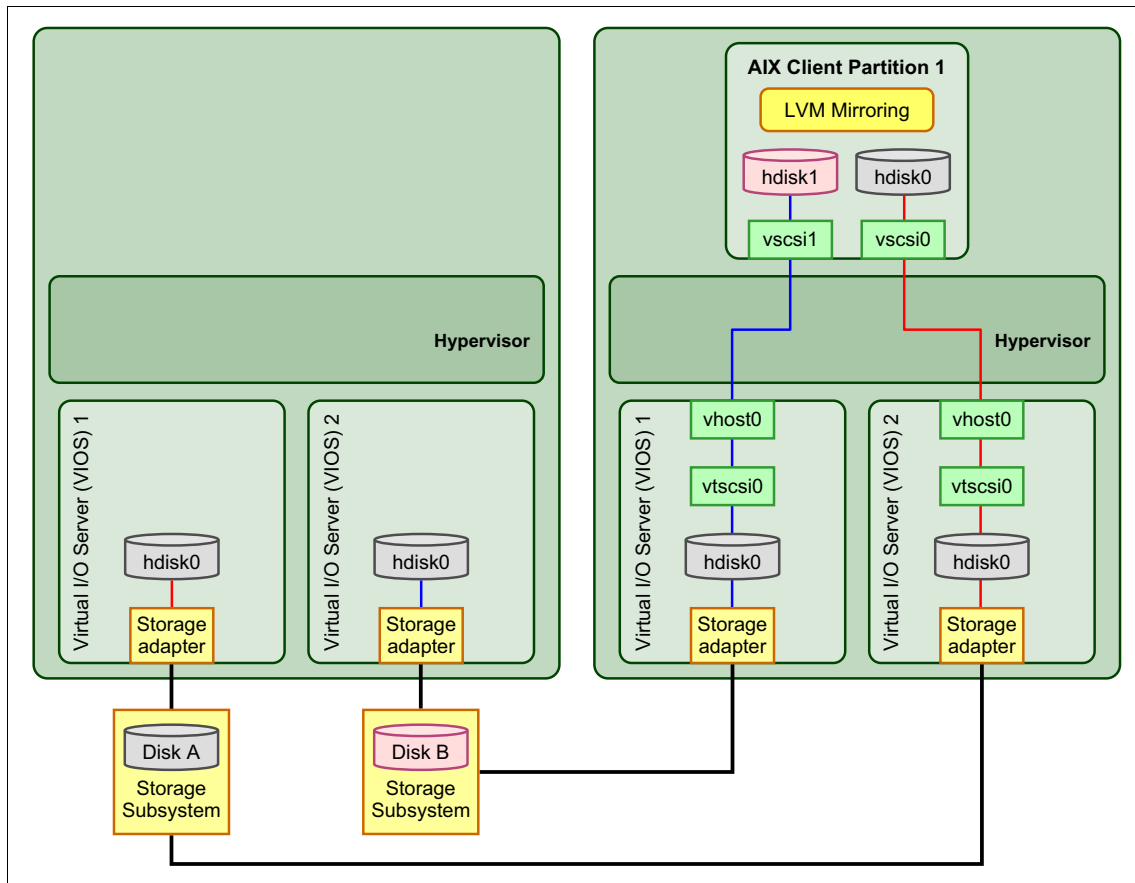


Figure 11-8 Dual VIOS and client mirroring to dual VIOS after migration

If the destination system has only one Virtual I/O Server, the migration is still possible and the same virtual SCSI setup is preserved at the client side. The destination Virtual I/O Server must have access to all disk spaces and the process creates two virtual SCSI adapters on the same Virtual I/O Server, as shown in Figure 11-9. In order to accomplish this, the user must specify the mapping.

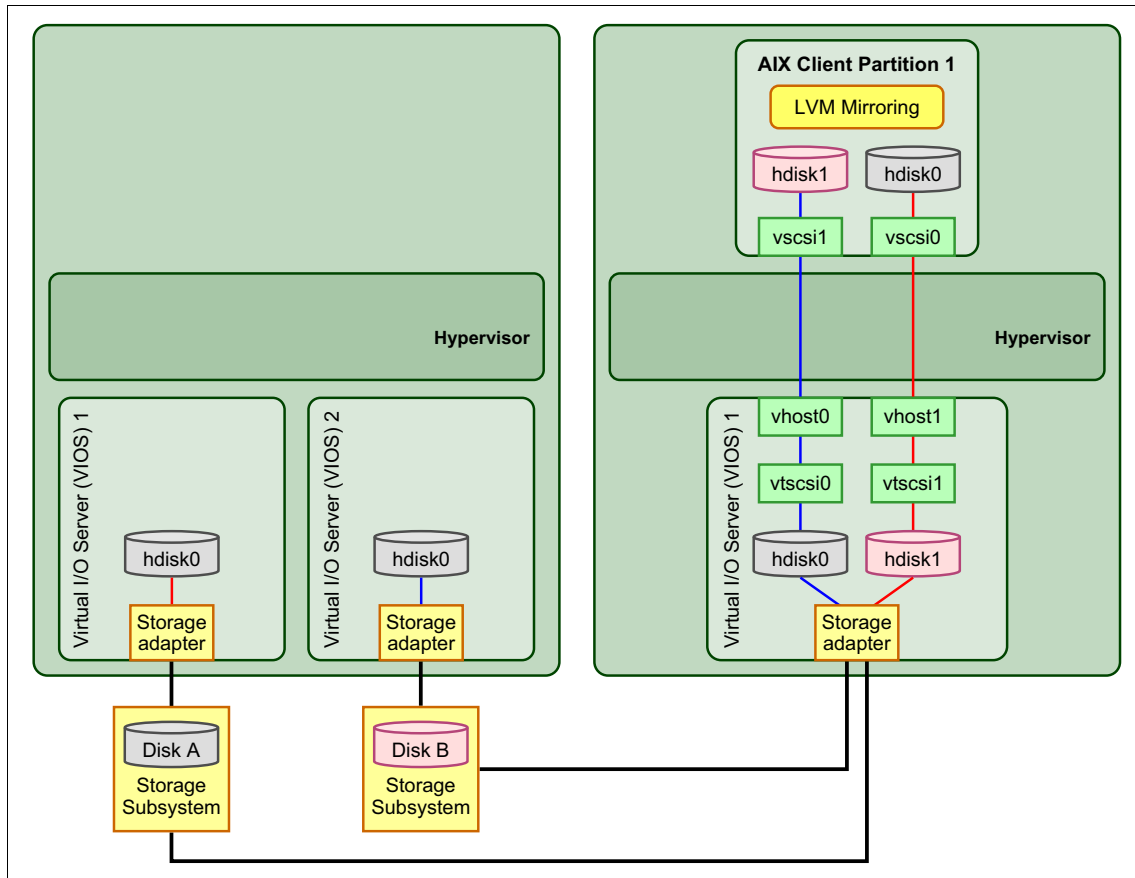


Figure 11-9 Dual VIOS and client mirroring to single VIOS after migration

Dual Virtual I/O Server and multipath I/O

With multipath I/O, the logical partition accesses the same disk data using two different paths, each provided by a separate Virtual I/O Server. One path is active and the other is standby.

The migration is possible only if the destination system is configured with two Virtual I/O Servers that can provide the same multipath setup. They both must have access to the shared disk data, as shown in Figure 11-10.

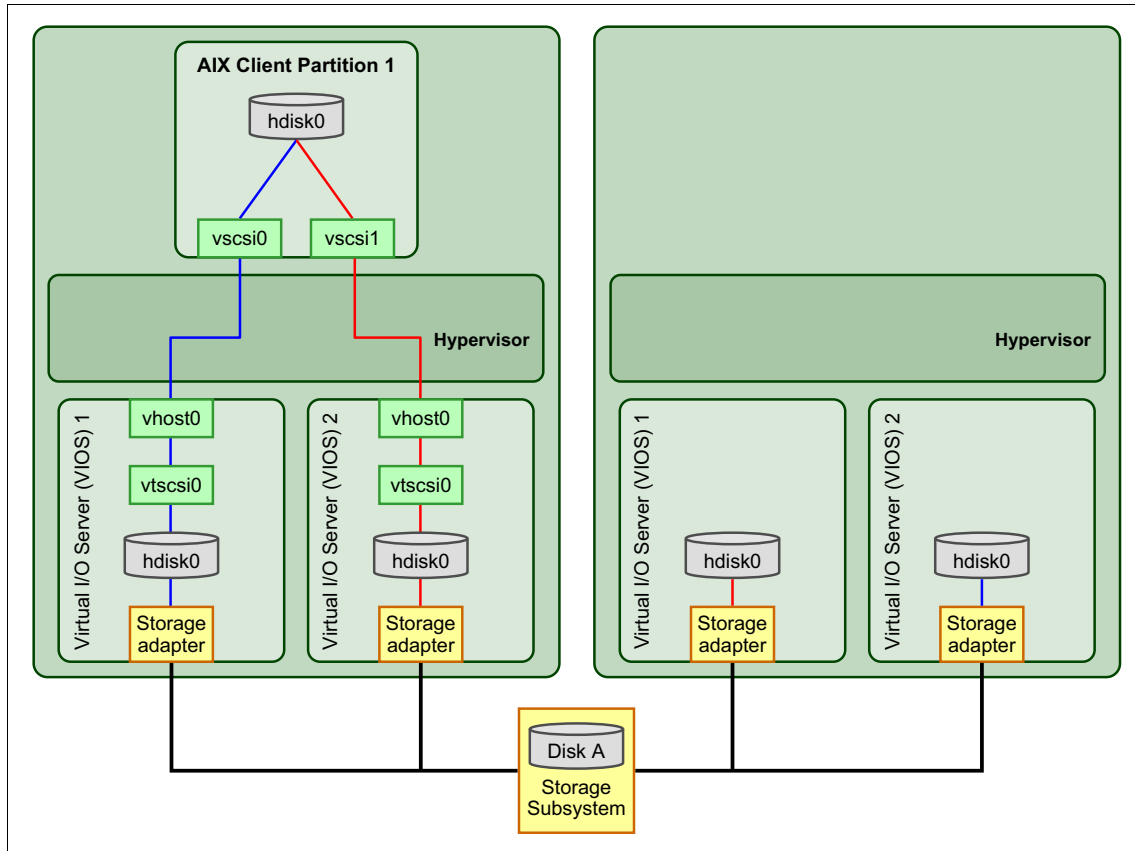


Figure 11-10 Dual VIOS and client multipath I/O to dual VIOS before migration

When migration is complete, on the destination system, the two Virtual I/O Servers are configured to provide the two paths to the data, as shown in Figure 11-11.

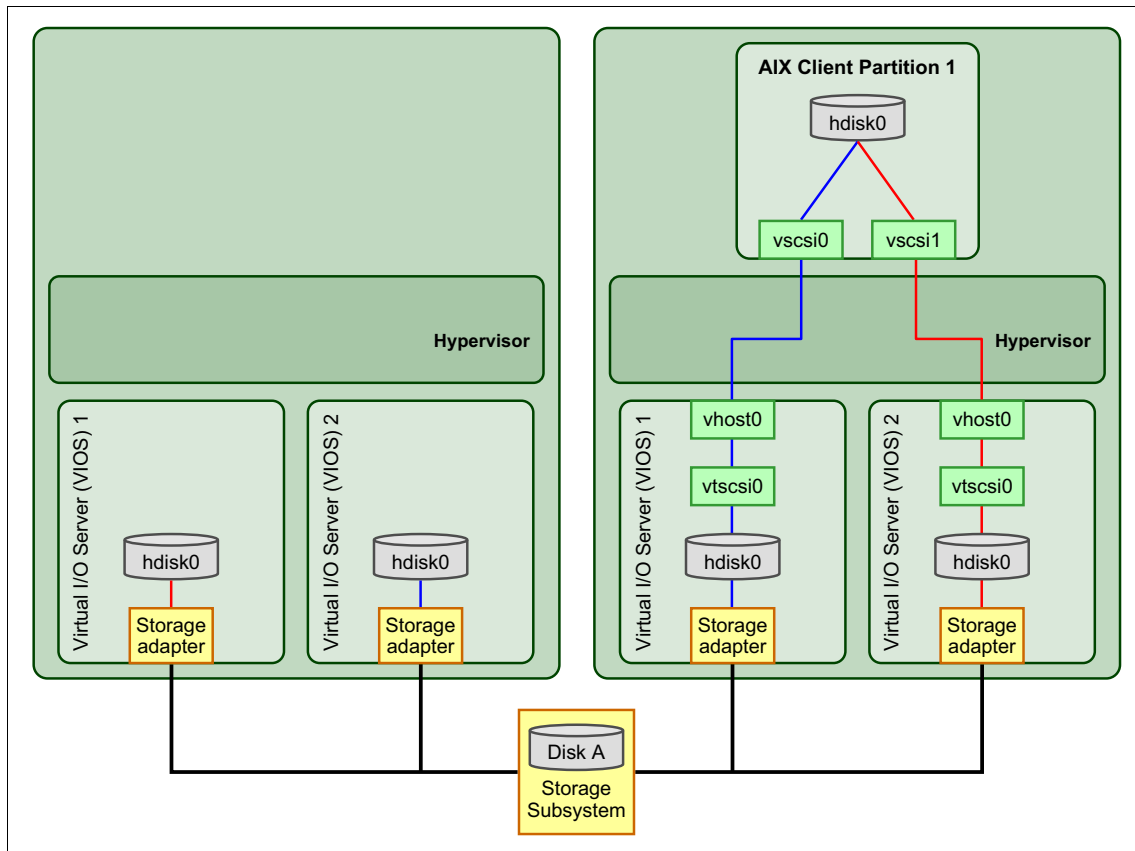


Figure 11-11 Dual VIOS and client multipath I/O to dual VIOS after migration

If the destination system is configured with only one Virtual I/O Server, the migration cannot be performed. The migration process would create two paths using the same Virtual I/O Server, but this setup is not allowed, because having two virtual target devices that map the same backing device on different virtual SCSI server devices is not possible.

To migrate the partition, you must first remove one path from the source configuration before starting the migration. The removal can be performed without interfering with the running applications. The configuration becomes a simple single Virtual I/O Server migration.

Single to dual Virtual I/O Server

A logical partition that is using only one Virtual I/O Server for virtual disks may be migrated to a system where multiple Virtual I/O Servers are available. Because the migration never changes a partition's configuration, only one Virtual I/O Server is used on the destination system.

If access to all disk data required by the partition is provided by only one Virtual I/O Server on the destination system, after migration the partition will use just that Virtual I/O Server. If no destination Virtual I/O Server provides all disk data, the migration cannot be performed.

When both destination Virtual I/O Servers have access to all the disk data, the migration can select either one or the other. When you start the migration, the HMC allows you to specify the destination Virtual I/O Server for each adapter. The HMC automatically makes a selection if you do not specify. The situation is shown in Figure 11-12.

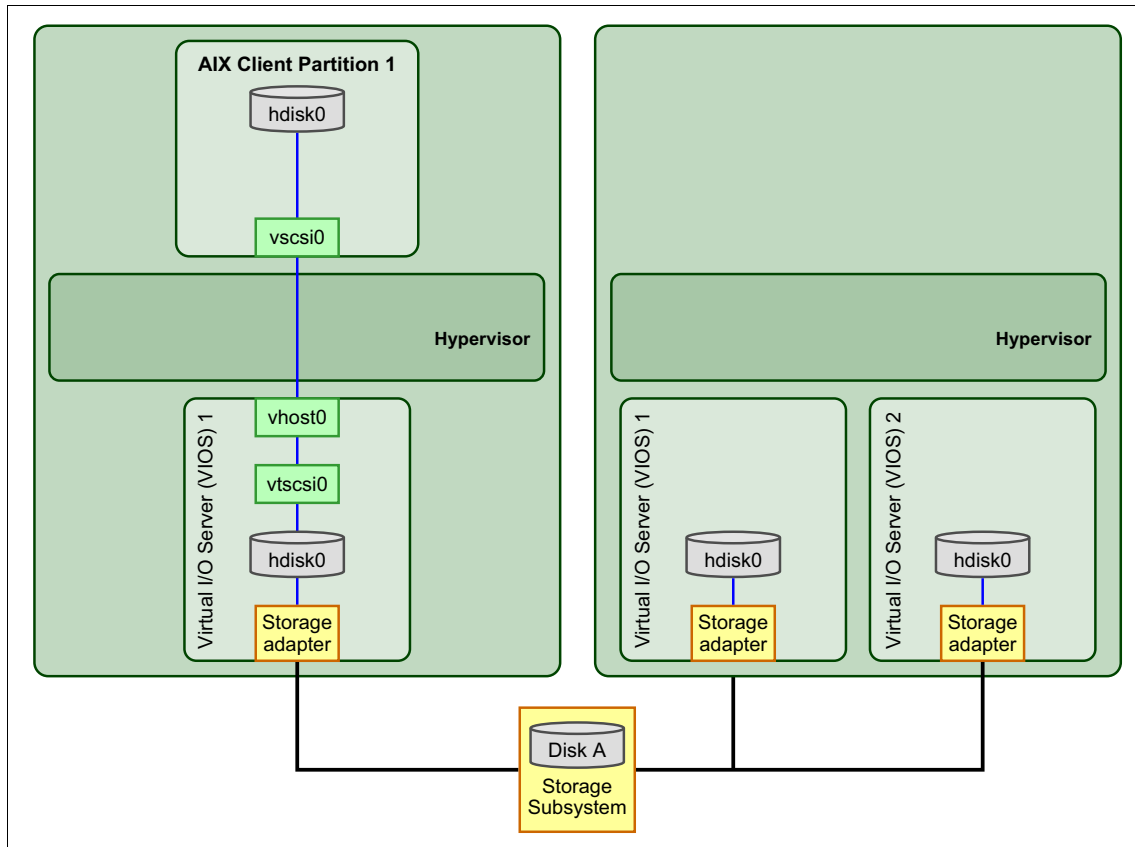


Figure 11-12 Single VIOS to dual VIOS before migration

When the migration is performed using the GUI on the HMC, a list of possible Virtual I/O Servers to pick from is provided. By default, the command-line interface makes the automatic selection if no specific option is provided.

After migration, the configuration is similar to the one shown in Figure 11-13.

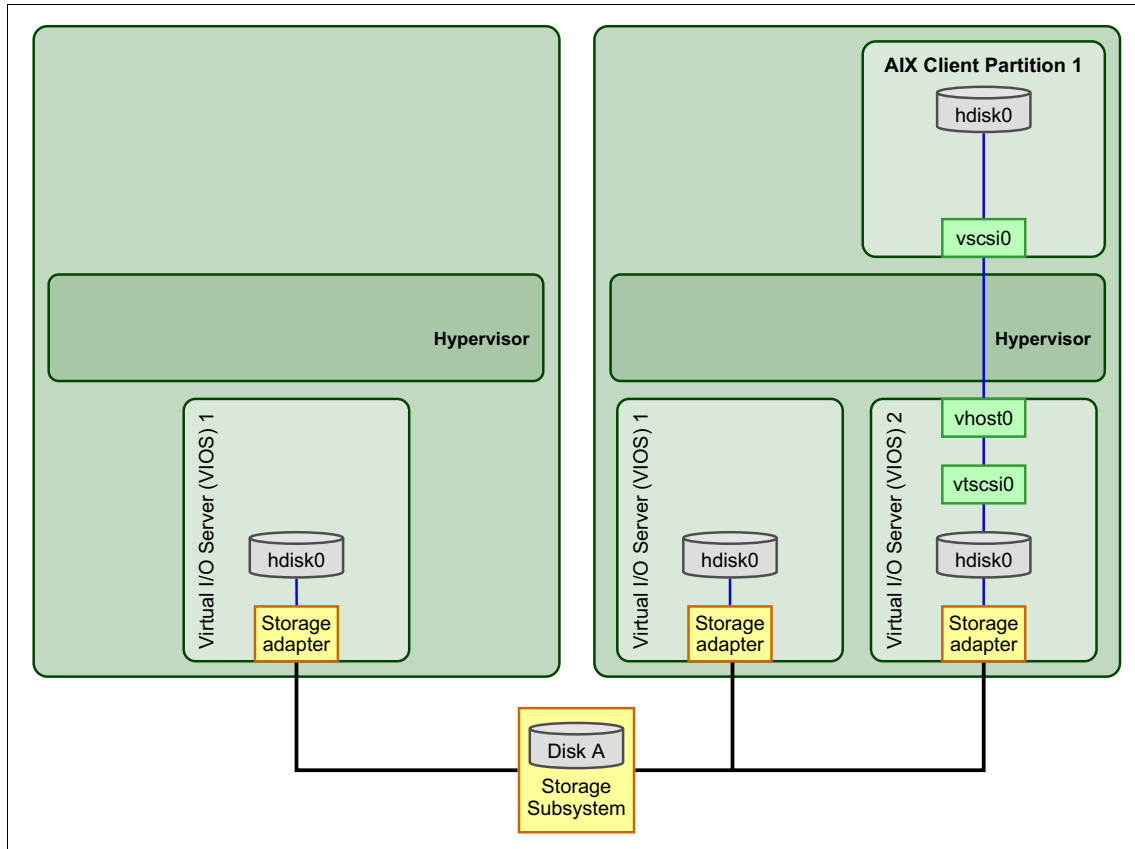


Figure 11-13 Single VIOS to dual VIOS after migration

11.2.7 Live Partition Mobility using Virtual Fibre Channel

Virtual Fibre Channel is a virtualization feature. Virtual Fibre Channel uses N_Port ID Virtualization (NPIV), and enables PowerVM logical partitions to access SAN resources using virtual Fibre Channel adapters mapped to a physical NPIV-capable adapter.

Figure 11-14 shows a basic configuration using virtual Fibre Channel and a single Virtual I/O Server in the source and destination systems before migration occurs.

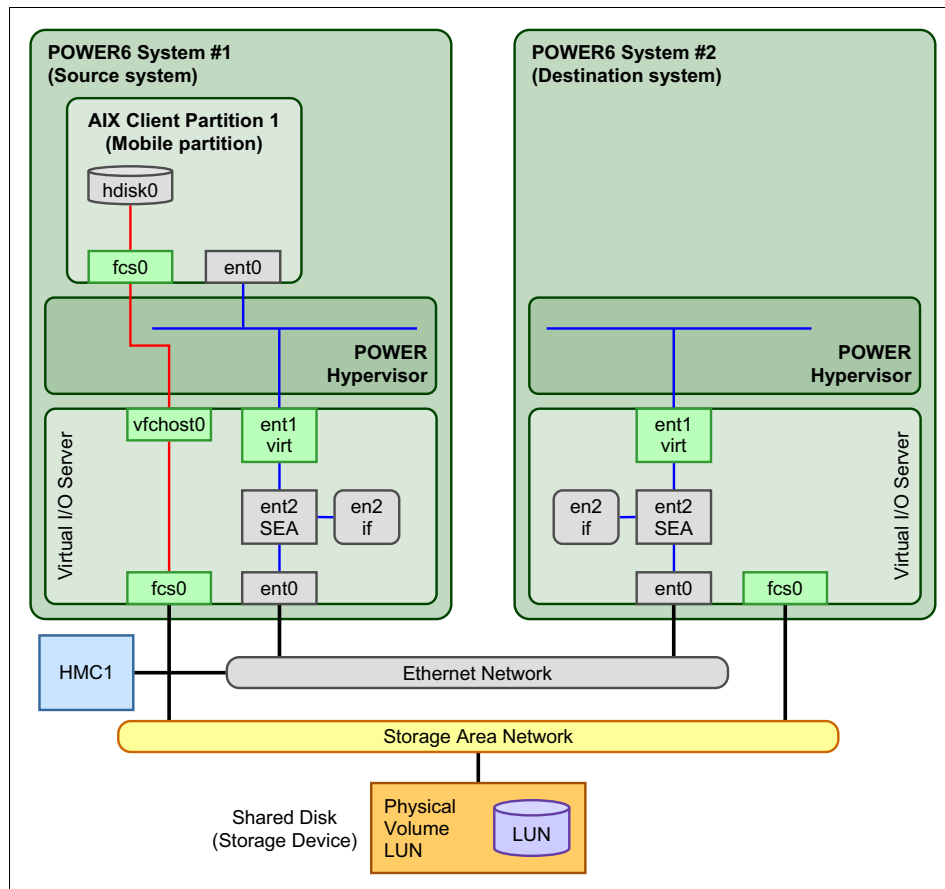


Figure 11-14 Basic virtual Fibre Channel infrastructure before migration

After migration, the configuration is similar to the one shown in Figure 11-15.

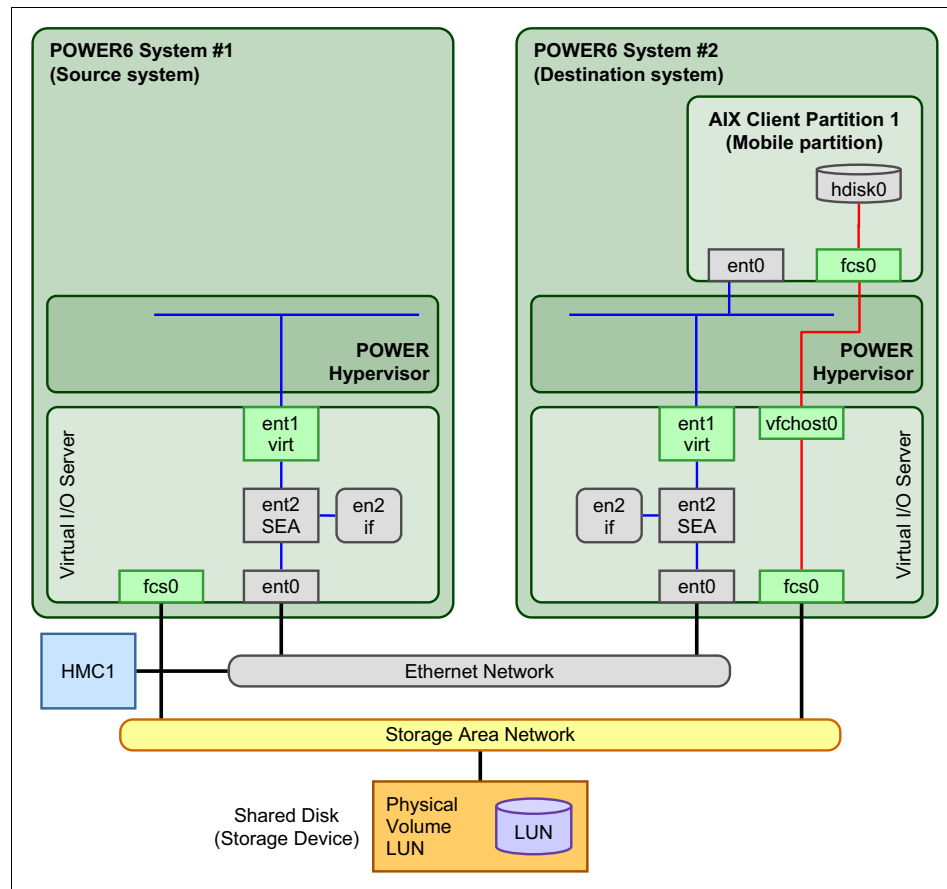


Figure 11-15 Basic virtual Fibre Channel infrastructure after migration

Benefits of virtual Fibre Channel

The addition of virtual Fibre Channel adapters reduces the number of components and steps necessary to configure shared storage in a Virtual I/O Server configuration:

- ▶ With virtual Fibre Channel support, you do not map individual disks in the Virtual I/O Server to the mobile partition. LUNs from the storage subsystem are zoned in a switch with the mobile partition's virtual Fibre Channel adapter using its worldwide port names (WWPNs), which greatly simplifies Virtual I/O Server storage management.

- ▶ LUNs assigned to the virtual Fibre Channel adapter appear in the mobile partition as standard disks from the storage subsystem. LUNs do not appear on the Virtual I/O Server unless the physical adapters WWPN is zoned.
- ▶ Standard multipathing software for the storage subsystem is installed on the mobile partition. Multipathing software is not installed into the Virtual I/O Server partition to manage virtual Fibre Channel disks. The absence of the software provides system administrators with familiar configuration commands and problem determination processes in the client partition.
- ▶ Partitions can take advantage of standard multipath features, such as load balancing across multiple virtual Fibre Channel adapters presented from dual Virtual I/O Servers.

Required components

The mobile partition must meet the requirements previously described, in addition, the following components must be configured in the environment:

- ▶ An NPIV-capable SAN switch
- ▶ An NPIV-capable physical Fibre Channel adapter on the source and destination Virtual I/O Servers
- ▶ HMC Version 7 Release 3.4, or later
- ▶ Virtual I/O Server Version 2.1 with Fix Pack 20.1, or later
- ▶ AIX 6.1 TL2 SP2, or later
- ▶ Red Hat Enterprise Linux 5.4 or later
- ▶ SuSE Linux Enterprise Server 10 SP3 or later
- ▶ Each virtual Fibre Channel adapter on the Virtual I/O Server mapped to an unique NPIV-capable physical Fibre Channel adapter. This means that you cannot have two or more virtual Fibre Channel adapters to the mobilizing client backed by a single physical Fibre Channel adapter.
- ▶ The destination Virtual I/O Server must have physical Fibre Channel adapters with `max_xfer_size` set greater than or equal to the source Virtual I/O Server physical Fibre Channel adapters.

The command to check it is:

```
lsattr -El fcs0 | grep max_xfer_size
```

- ▶ Each virtual Fibre Channel adapter on the mobile partition mapped to a virtual Fibre Channel adapter in the Virtual I/O Server
- ▶ At least one LUN mapped to the mobile partition's virtual Fibre Channel adapter

Dual Virtual I/O Server and virtual Fibre Channel multipathing

With multipath I/O, the logical partition accesses the same storage data using two different paths, each provided by a separate Virtual I/O Server.

Note: With NPIV-based disks, both paths can be active. For NPIV and virtual Fibre Channel, the storage multipath code is loaded into the mobile partition. The multipath capabilities depend on the storage subsystem type and multipath code deployed in the mobile partition.

The migration is possible only if the destination system is configured with two Virtual I/O Servers that can provide the same multipath setup. They both must have access to the shared disk data, as shown in Figure 11-16.

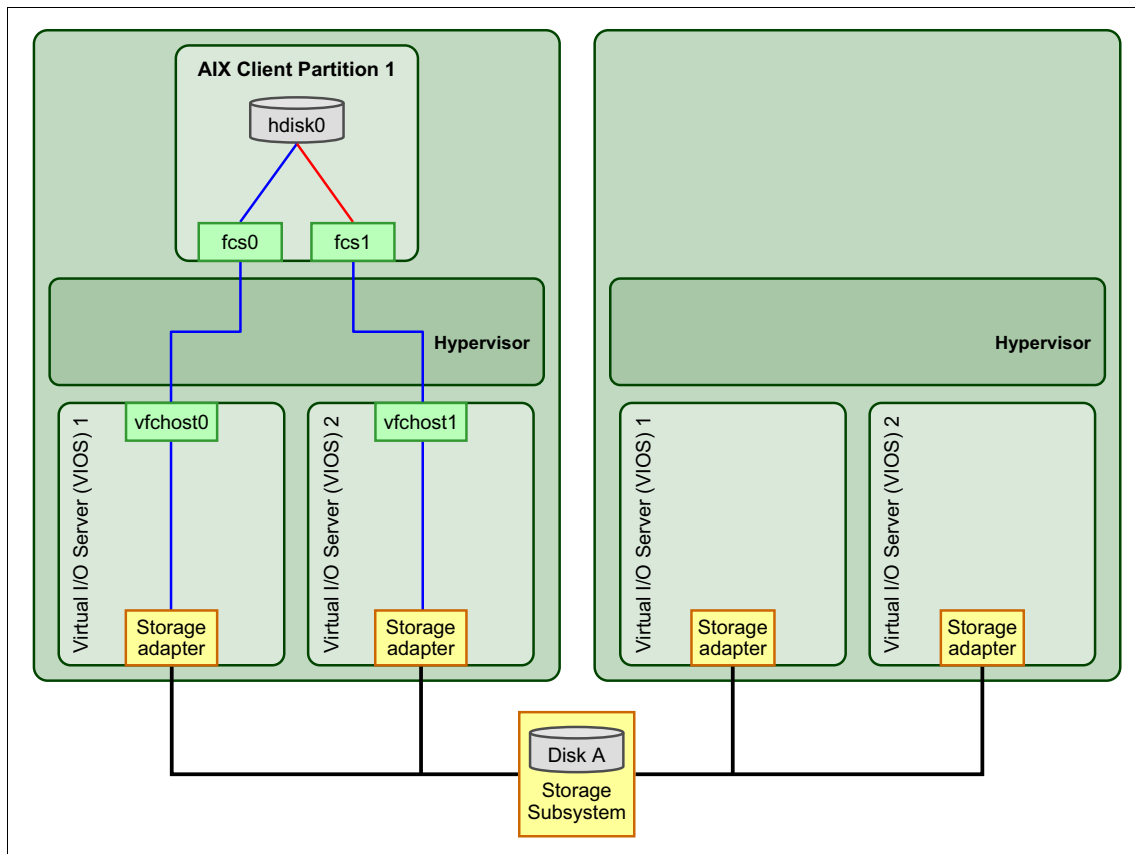


Figure 11-16 Dual VIOS and client multipath I/O to dual FC before migration

When migration is complete, on the destination system, the two Virtual I/O Servers are configured to provide the two paths to the data, as shown in Figure 11-17.

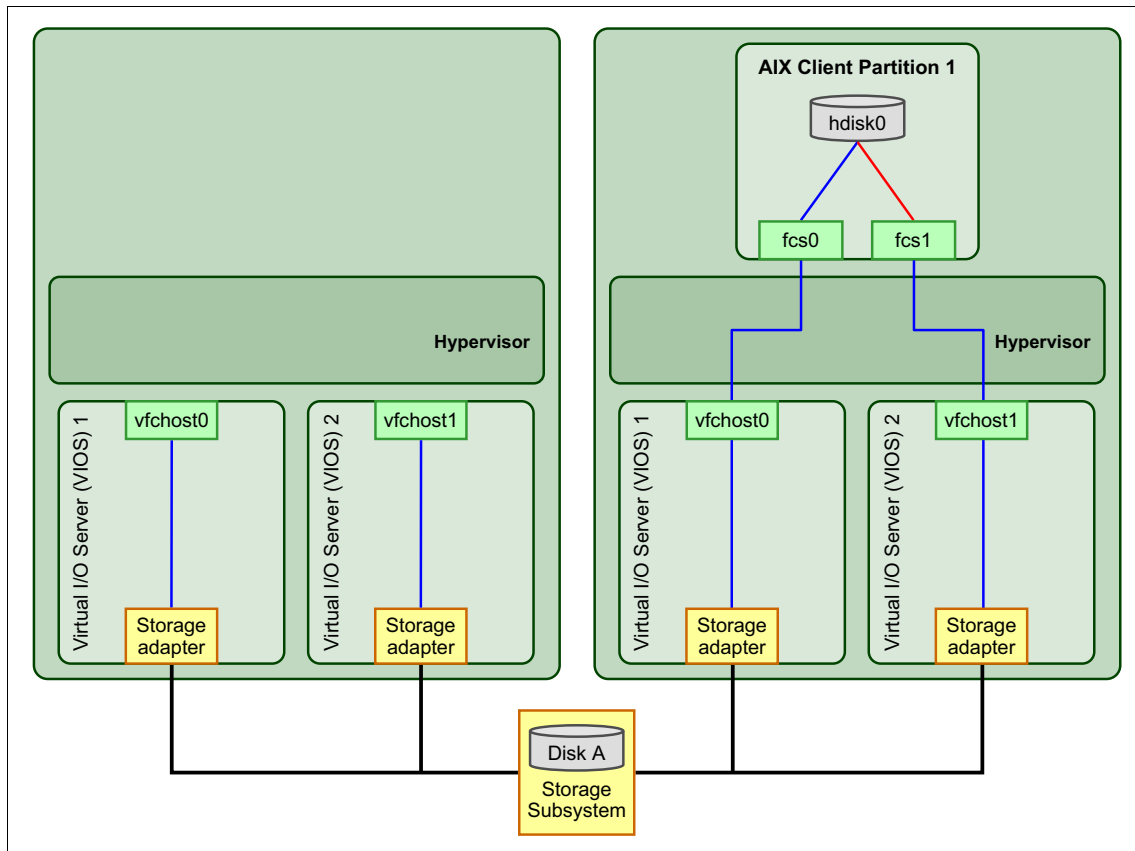


Figure 11-17 Dual VIOS and client multipath I/O to dual VIOS after migration

If the destination system is configured with only one Virtual I/O Server, the migration cannot be performed. The migration process would create two paths using the same Virtual I/O Server, but this setup of having one virtual Fibre Channel host device mapping the same LUNs on different virtual Fibre Channel adapters is not recommended.

To migrate the partition, you must first remove one path from the source configuration before starting the migration. The removal can be performed without interfering with the running applications. The configuration becomes a simple single Virtual I/O Server migration.

11.2.8 Multiple concurrent migrations

Important: PowerVM brings significant improvements to the Live Partition Mobility (LPM), mainly to performance and to the multiple migrations. IBM PowerVM and the Hardware Management Console now support up to 16 active, inactive, and suspended migrations simultaneously. Improvements in single-session LPM performance can accelerate mobility for a single session by up to 3X over previous releases. Note, IVM managed systems support up to eight migrations simultaneously.

The same system can handle multiple concurrent partition migrations, any mix of either inactive, active, or suspended.

In many scenarios, more than one migration may be started on the same system. For example:

- ▶ A review of the entire infrastructure detects that a different system location of some logical partition may improve global system usage and service quality.
- ▶ A system is planned to enter maintenance and must be shut down. Some of its partitions cannot be stopped or the planned maintenance time is too long to satisfy service level agreements.

The maximum number of concurrent migrations on a system can be identified by using the **lslparmigr** command on the HMC, with the following syntax:

```
lslparmigr -r sys -m <system>
```

Example:

```
hscroot@hmc8:~> lslparmigr -r sys -m p740 | sed "s/,/\n/g"
inactive_lpar_mobility_capable=1
num_inactive_migrations_supported=4
num_inactive_migrations_in_progress=0
active_lpar_mobility_capable=1
num_active_migrations_supported=16
num_active_migrations_in_progress=0
inactive_prof_policy=config
sys_firmware_num_inactive_migrations_supported=4
sys_firmware_num_active_migrations_supported=16
os400_lpar_mobility_capable=1
hscroot@hmc8:~>
```

Several practical considerations should be taken into account when planning for multiple migrations, especially when the time required by the migration process has to be evaluated.

For each mobile partition, you must use an HMC GUI wizard or an HMC command. While a migration is in progress, you can start another one. When the number of migrations to be executed grows, the setup time using the GUI can become long and you should consider using the CLI instead. The **migr1par** command may be used in scripts to start multiple migrations in parallel. Starting in HMC V7R7.6, the **migr1par** command was expanded to allow the user to specify multiple partitions to migrate.

An active migration requires more time to complete than an inactive migration because the system performs additional activities to keep applications running while the migration is in progress.

Consider the following information:

- ▶ The time required to complete an active migration depends on the size of the memory to be migrated and on the mobile partition's workload.
- ▶ The Virtual I/O Servers selected as mover service partitions are loaded by memory moves and network data transfer, as follows:
 - High speed network transfers can become processor-intensive workloads.
 - At a minimum, four concurrent (active, inactive, or suspended) migrations can be managed by the same mover service partition with the latest Virtual I/O Server Version 2.2.2.0 supporting eight concurrent migrations.

The active migration process has been designed to handle any partition memory size and it is capable of managing any memory workload. Applications can update memory with no restriction during migration and all memory changes are taken into account, so elapsed migration time can change with workload. Although the algorithm is efficient, planning the migration during low activity periods can help to reduce migration time.

Virtual I/O Servers selected as mover service partitions are involved in partition's memory migration and must manage high network traffic. Network management can cause high CPU usage and usual performance considerations apply; use uncapped Virtual I/O Servers and add virtual processors if the load increases. Alternatively, create dedicated Virtual I/O Servers on the source and destination systems that provide the mover service function separating the service network traffic from the migration network traffic. You can combine or separate virtualization functions and mover service functions to suit your requirements.

If multiple mover service partitions are available on either the source or destination systems, we suggest distributing the load among them. This process can be done explicitly by selecting the mover service partitions, either by using the GUI or the CLI. Each mover service partition can manage at a minimum four concurrent (active, inactive, or suspended) migrations with the latest Virtual I/O

Server 2.2.2.0 supporting eight concurrent migrations and explicitly using multiple Virtual I/O Servers avoids queuing of requests.

When running eight concurrent migrations through a Virtual I/O Server, use a 10 Gbps network.

Tip: If there are multiple network connections between your Virtual I/O Server partitions, test the line speed prior to the migration, which allows you to specify the fastest connection to use during the migration.

11.2.9 Remote Live Partition Mobility

This section focuses on Live Partition Mobility and its ability to migrate a logical partition between two IBM Power Systems servers each managed by a separate Hardware Management Console. Remote migrations require coordinated movement of a partition's state and resources over a secure network channel to a remote HMC. The following information talks about HMC, but most of the information pertains to IVM managed systems as well. Do note that migrations between IVM and HMC managed systems are not supported.

The following list indicates the high-level prerequisites for remote migration. If any of the following elements are missing, a migration cannot occur:

- ▶ A ready source system that is migration-capable.
- ▶ A ready destination system that is migration-capable.
- ▶ Compatibility between the source and destination systems.
- ▶ Destination system managed by a remote HMC.
- ▶ Network communication between local and remote HMC.
- ▶ A migratable, ready partition to be moved from the source system to the destination system. For an inactive migration, the partition must be turn off, but must be capable of booting on the destination system.
- ▶ For active and suspended migrations, an MSP on the source and destination systems.
- ▶ One or more SANs that provide connectivity to all of the mobile partition's disks to the Virtual I/O Server partitions on both the source and destination servers. The mobile partition accesses all migratable disks through devices (virtual Fibre Channel, virtual SCSI, or both). The LUNs used for virtual SCSI must be zoned and masked to the Virtual I/O Servers on both systems. Virtual Fibre Channel LUNs should be configured as described *IBM PowerVM Virtualization Managing and Monitoring*, SG24-7590.

- ▶ Hardware-based iSCSI connectivity can be used in addition to SAN. SCSI reservation must be disabled. IBM does not support virtualization of iSCSI LUNs using SW initiator. That is, an hdisk that is created on the VIOS as a result of iSCSI SW initiator attachment cannot be mapped to a client as a vSCSI disk on the client. However, we do continue to support virtualizing iSCSI LUNs using the TCP/IP Offload Engine (TOE) adapter, even though we stopped shipping it. The storage tested is IBM System Storage N series attached through the TOE adapter. An alternative recommendation is setting up a Shared Ethernet Adapter (SEA) on the VIOS and then running the iSCSI SW initiator on the client instead of in the VIOS.
- ▶ The mobile partition's virtual disks must be mapped to LUNs; they cannot be part of a storage pool or logical volume on the Virtual I/O Server.
- ▶ One or more physical IP networks (LAN) that provide the necessary network connectivity for the mobile partition through the Virtual I/O Server partitions on both the source and destination servers. The mobile partition accesses all migratable network interfaces through virtual Ethernet devices.
- ▶ An RMC connection to manage inter-system communication

Remote migration operations require that each HMC has RMC connections to its individual system's Virtual I/O Servers and a connection to its system's service processors. The HMC does not have to be connected to the remote system's RMC connections to its Virtual I/O Servers nor does it have to connect to the remote system's service processor.

The remote active, inactive, and suspended migrations follow the same workflow as described previously. The local HMC, which manages the source server in a remote migration, serves as the controlling HMC. The remote HMC, which manages the destination server, receives requests from the local HMC and sends responses over a secure network channel.

Requirements for remote migration

This feature allows a user to migrate a client partition to a destination server that is managed by a different HMC. The function relies on Secure Shell (SSH) to communicate with the remote HMC.

The following list indicates the requirements for remote HMC migrations:

- ▶ A local HMC managing the source server
- ▶ A remote HMC managing the destination server
- ▶ Network access to a remote HMC
- ▶ SSH key authentication to the remote HMC

To initiate the remote migration operation, you may use only the HMC that contains the mobile partition.

The steps to configure Virtual I/O Servers, client partition, mover service partitions, and partition profiles do not change.

Use dedicated networks with 1 Gbps bandwidth, or more. This applies for each involved HMC, Virtual I/O Server, and mover service partition.

A recommendation is to have some IBM Power Systems servers use private networks to access the HMC. The ability to migrate a partition remotely allows Live Partition Mobility between systems managed by HMCs that are also using separate private networks.

Figure 11-18 displays the Live Partition Mobility infrastructure involving the two remote HMCs and their respective managed systems.

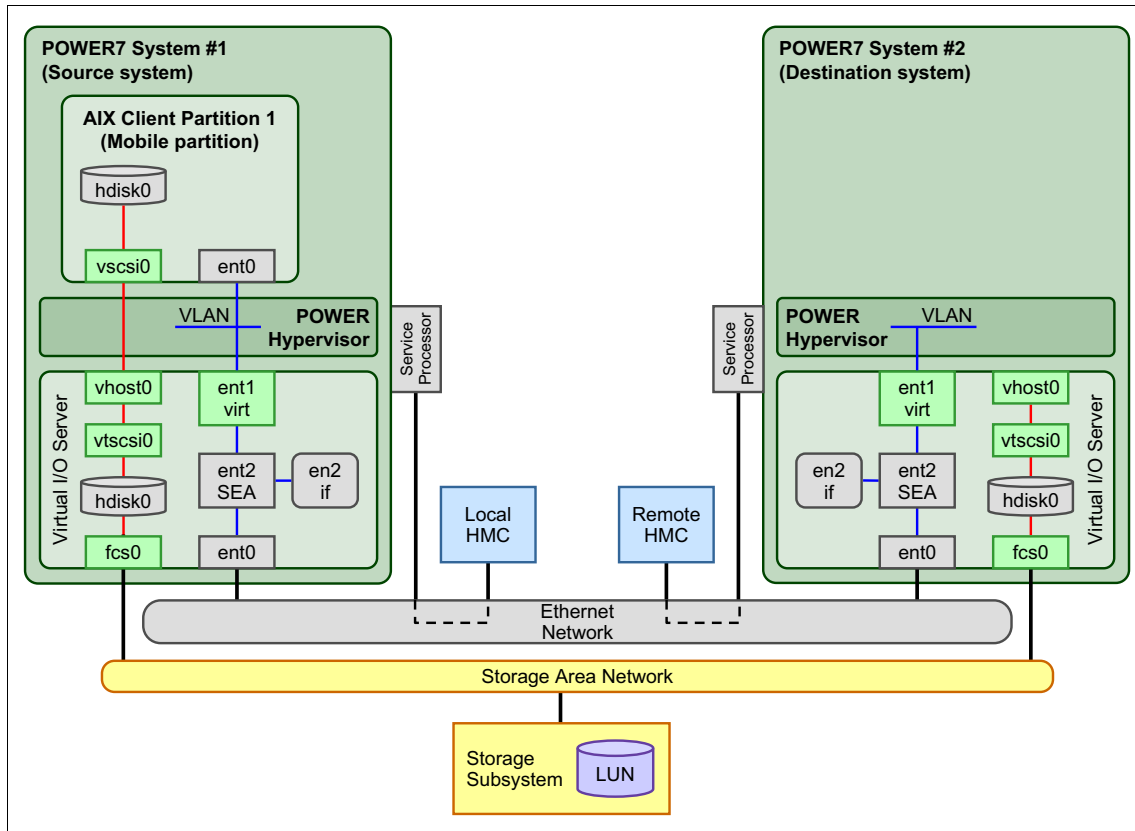


Figure 11-18 Live Partition Mobility infrastructure with two HMCs

Figure 11-19 displays the infrastructure involving private networks that link each service processor to its HMC. The HMC for both systems contains a second network interface that is connected to the public network.

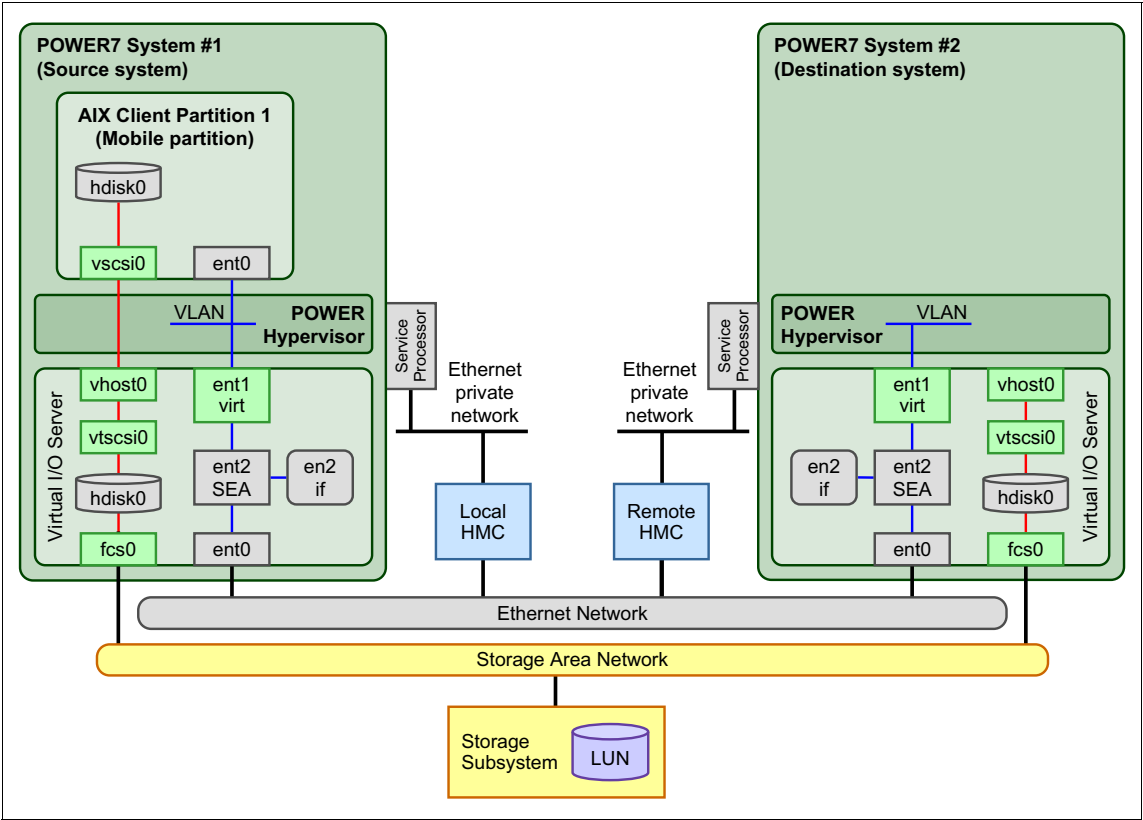


Figure 11-19 Live Partition Mobility infrastructure using private networks

Figure 11-20 shows the situation where one POWER System is in communication with the HMC on a private network, and the destination server is communicating by using the public network.

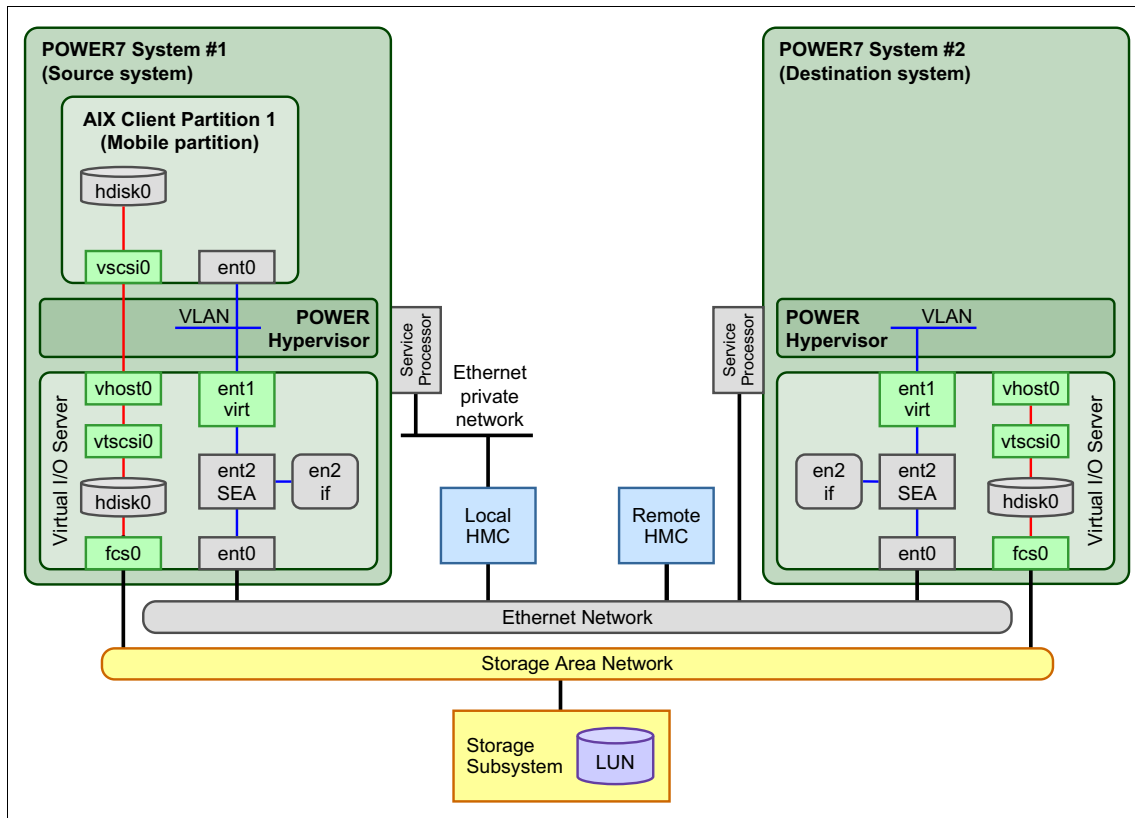


Figure 11-20 One public and one private network migration infrastructure

11.2.10 Processor compatibility modes

Regarding the Processor compatibility modes, see Appendix B, “POWER processor modes” on page 681.

For additional information about live partition mobility, see:

<http://pic.dhe.ibm.com/infocenter/powersys/v3r1m5/index.jsp?topic=%2Fp7hc3%2Fiphc3kickoff.htm>

For information about Processor Compatibility specifics, see:

<http://pic.dhe.ibm.com/infocenter/powersys/v3r1m5/topic/p7hc3/iphc3pcm.htm>

11.2.11 General considerations

This section is dedicated to concentrate the relevant considerations related to the topics that were discussed previously.

Storage pool usage

Because the mobile partition's external disk space must be available to the Virtual I/O Servers on the source and destination systems, you cannot use storage pools. Each Virtual I/O Server must create virtual target devices using physical disks and not logical volumes.

Performance considerations

Active partition migration involves moving the state of a partition from one system to another while the partition is still running. The mover service partitions working with the hypervisor use partition virtual memory functions to track changes to partition memory state on the source system while it is transferring memory state to the destination system.

During the migration phase, an initial transfer of the mobile partition's physical memory from the source to the destination occurs. Because the mobile partition is still active, a portion of the partition's resident memory will almost certainly have changed during this pass. The hypervisor keeps track of these changed pages for retransmission to the destination system in a dirty page list. It makes additional passes through the changed pages until the mover service partitions detects that a sufficient amount of pages are clean or the timeout is reached.

The speed and load of the network that is used to transfer state between the source and destination systems influence the time required for both the transfer of the partition state and the performance of any remote paging operations.

The amount of changed resident memory after the first pass is controlled more by write activity of the hosted applications than by the total partition memory size. Nevertheless, a reasonable assumption is that partitions with a large memory requirement have higher numbers of changed resident pages than smaller ones.

To ensure that active partition migrations are truly nondisruptive, even for large partitions, the POWER Hypervisor resumes the partition on the destination system before all the dirty pages have been migrated over to the destination. If the mobile partition tries to access a dirty page that has not yet been migrated from the source system, the hypervisor on the destination sends a demand paging request to the hypervisor on the source to fetch the required page.

Providing a high-performance network between the source and destination mover partitions and reducing the partition's memory update activity prior to migration will improve the latency of the state transfer phase of migration. We

suggest using a dedicated network for state transfer, with a nominal bandwidth of at least 1 Gbps.

AIX migration

An AIX partition continues running during an active migration. Most AIX features work seamlessly before, during, and after the migration. These include, but are not limited to, the following features:

- ▶ System and advanced accounting
- ▶ Workload manager
- ▶ System trace
- ▶ Resource sets:
Including exclusive-use processor resource sets
- ▶ Pinned memory
- ▶ Large memory pages:
Huge memory pages cannot be used

Performance monitoring tools (such as commands **topas**, **tprof**, **filemon**, and so on) can run on a mobile partition during an active migration. However, the data that these tools report during the migration process might not be significant, because of underlying hardware changes, performance monitor counters that may be reset, and so on.

Although AIX is migration safe, verify that any applications you are running are migration safe or aware.

For information about application aware migrations, see *IBM PowerVM Virtualization Managing and Monitoring*, SG24-7590, which is available at:

<http://www.redbooks.ibm.com/redbooks/pdfs/sg247590.pdf>

IBM i migration

In general, applications and operating system are unaware that the IBM i partition is moved from one system to another.

There are some exceptions to this:

- ▶ Applications that recognize:
 - System serial number
 - LPAR ID
 - System type/model

Other values may not be correct until the existing configuration is deleted and recreated.

- Operations Console configuration for a cleared Service Tool LAN adapter is disabled. The operations console configuration must be deleted and recreated to re-configure a Service Tool LAN adapter that has its TCP/IP settings cleared.

Linux migration

A Linux partition continues running during an active migration. Many features on supported Linux operating systems work seamlessly before, during, and after migration, such as IBM RAS tools and dynamic reconfiguration.

Similar to AIX, Linux is migration-safe. A good idea is to verify that any applications not included in the full distributions of the supported Linux operating systems are migration-safe or aware.

For information about application aware migrations, see *IBM PowerVM Virtualization Managing and Monitoring*, SG24-7590, which is available at:

<http://www.redbooks.ibm.com/redbooks/pdfs/sg247590.pdf>

Distance considerations

There are no architected maximum distances between systems for Live Partition Mobility. The maximum distance is dictated by the network and storage configuration used by the systems. Provided both systems are on the same network, are connected to the same shared storage, and are managed by the same HMC, then Live Partition Mobility will work. Standard long-range network and storage performance considerations apply.

11.3 Suspend and Resume planning

This section covers planning details to use the Suspend and Resume capability in a PowerVM Standard or Enterprise Edition environment.

11.3.1 Configuration requirements

You can suspend a logical partition only when the logical partition is capable of suspension. At this time of writing, the Suspend/Resume capability requires the following minimum firmware and software levels:

- ▶ POWER 7 firmware 7.2.0 SP1
- ▶ PowerVM Standard Edition

- ▶ HMC V7R7.2.0
- ▶ VIOS 2.2.0.11-FP24 SP01
- ▶ AIX 7.1 TL0 SP2 or AIX 6.1 TL6 SP3
- ▶ IBM i 7.1 TR2 with HMC V7R7.3.0, VIOS 2.2.0.12-FP24 SP02, and POWER7 firmware Ax730_xxx

The maximum supported concurrent operations for Suspend/Resume is limited to 4 – unlike concurrent active partition mobility operations for which this limited got increased. However, there is no limitation for the maximum number of partitions that can be in a suspended state.

Note: The partition requirements for Live Partition Mobility also apply for suspending partitions.

The configuration requirements for suspending a logical partition are as follows:

- ▶ The reserved storage device must be kept persistently associated with the logical partition.
- ▶ The HMC ensures that the Reserved Storage Device Pool is configured with at least one active VIOS partition available in the pool.
- ▶ The logical partition's property "Allow this partition to be suspended." must be set (for an IBM i partition this can only be changed when the partition is deactivated).
- ▶ The logical partition must not have physical I/O adapters assigned.
- ▶ The logical partition must not be a full system partition, a VIOS partition or a service partition.
- ▶ The logical partition must not be an alternative error logging partition.
- ▶ The logical partition must not have a Barrier Synchronization Register (BSR).
- ▶ The logical partition must not have huge memory pages.
- ▶ The logical partition must not have its *rootvg* volume group on a logical volume, on a file-backed device, on a shared storage pool device, or have any exported optical drives.
- ▶ The logical partition must not have a virtual SCSI optical or tape device assigned to the logical partition.
- ▶ Monitoring systems must be manually stopped/resumed while suspending and resuming logical partitions.
- ▶ Both WWPNs of a virtual Fibre Channel adapter must be zoned in the switch.
- ▶ A Dynamic Platform Optimizer (DPO) operation must not be running.

- ▶ When the logical partition is in the suspend state, you must not perform any operation that changes the state of the logical partition properties.
- ▶ The following restrictions additionally apply for an IBM i partition enabled for suspension:
 - You cannot suspend an IBM i logical partition while it is *active* in a cluster.
 - You cannot activate the logical partition with a partition profile which has a virtual SCSI server adapter.
 - You cannot activate the logical partition with a partition profile which has a virtual SCSI client adapter that is hosted by another IBM i logical partition.
 - You cannot dynamically add any virtual SCSI server adapter.
 - You cannot dynamically add any virtual SCSI client adapter that is hosted by another IBM i logical partition.
 - You cannot dynamically add any physical I/O adapters.
 - You cannot suspend an IBM i logical partition with a varied on NPIV attached tape device.
 - All IBM i virtual disks must be backed by physical volumes.

11.3.2 The Reserved Storage Device Pool

The partition state is stored on a persistent storage device that must be assigned on Reserved Storage Device Pool interface at the HMC. The Reserved Storage Device Pool has the assigned storage devices to save data for partitions. The storage device space required is approximately 110% of the partition's configured maximum memory size.

A Reserved Storage Device Pool has reserved storage devices called *paging space devices* and it is basically like a Shared Memory Pool of memory size 0. Paging space on a storage device is required for each partition to be suspended.

One Virtual I/O Server must be associated as the Paging Service Partition to the Reserved Storage Device Pool. Additionally, you can associate a second Virtual I/O Server partition with the Reserved Storage Device Pool in order to provide a redundant path and therefore higher availability to the paging space devices.

The Reserved Storage Device Pool is visible on HMC and can be accessed only when the Hypervisor is suspend capable. You can access the Reserved Storage Pool through the HMC CLI and GUI interfaces.

During a suspend operation, the HMC assigns a storage device from Reserved Storage Device Pool and it automatically picks an unused and suitable (size suggested by Hypervisor) device from this pool to store partition suspend data. Reserved storage device must be available in the Reserved Storage Device Pool at the time of suspending a logical partition.

Notes:

SAN disks typically have better performance than local disks for Suspend/Resume operations when allocated to the Reserved Storage Device Pool.

Using a Virtual I/O Server's Shared Storage Pool for creating paging space devices for Suspend/Resume or Active Memory Sharing (AMS) is not supported.

Next we illustrate how paging devices are allocated from the Reserved Storage Device Pool.

In this example, partitions 1, 2, and 3 use paging space devices 1, 2, and 3, which are SAN disks. Partition 4 uses paging space device 4, which is a local disk assigned to Paging VIOS partition 2. Both VIOS partitions are connected to the SAN as illustrated by the black lines. Green lines indicate the paging space devices mapped by the Paging VIOS partition 1, and blue lines indicate paging space devices mapped by the Paging VIOS Partition 2. Paging space devices 2 and 3 have redundant paths but not paging device 1.

See Figure 11-21 for a diagram of these concepts.

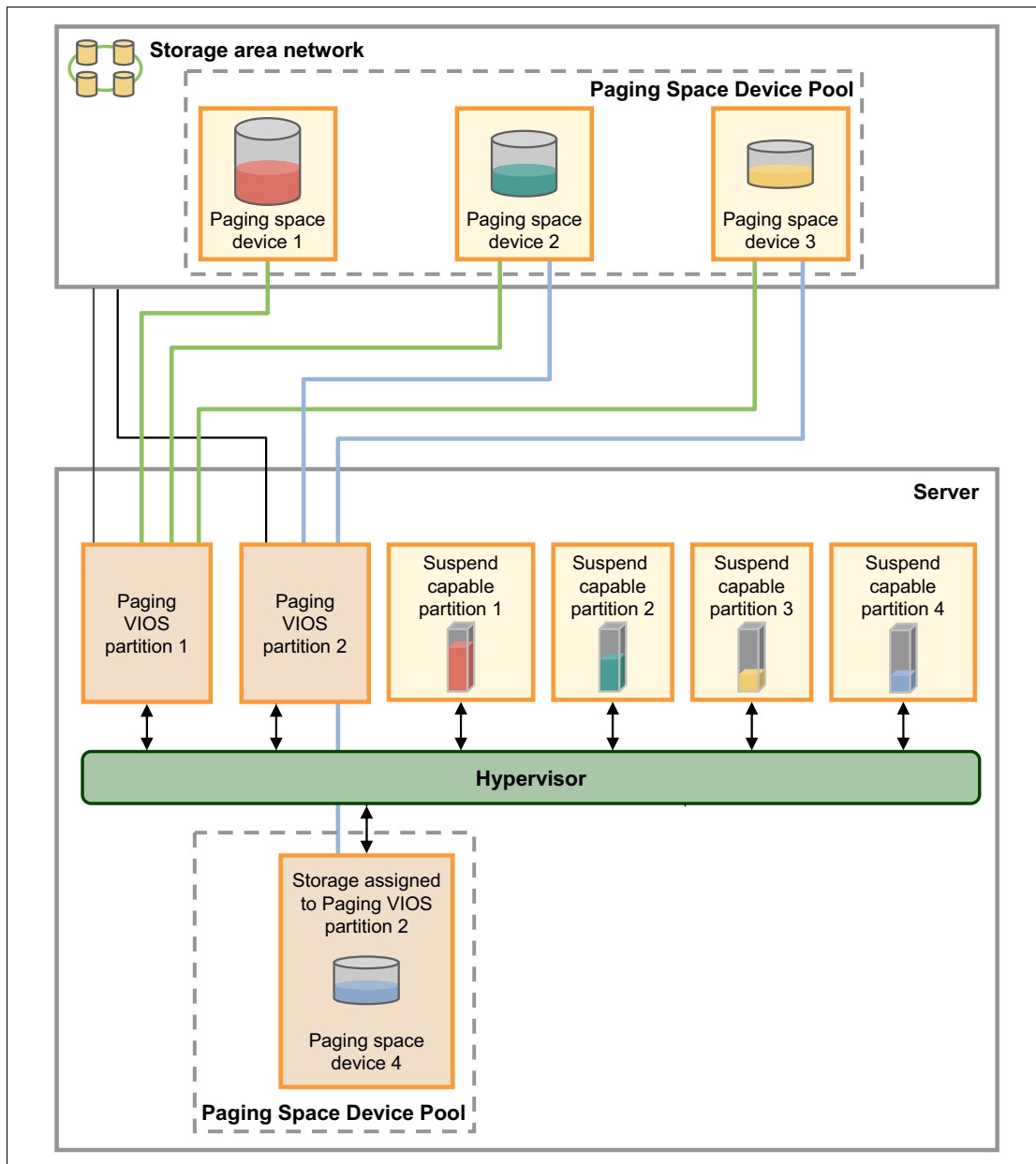


Figure 11-21 Reserved Storage Device Pool

On the PowerVM Standard Edition, the Reserved Storage pool interface is used to manage paging spaces in the pool. You can perform the following operations on the Reserved Storage Pool interface:

- ▶ Create/Delete the Reserved Storage Device Pool
- ▶ Add/Remove VIOS to/from the pool
- ▶ Add/Remove reserved storage devices to/from the pool

11.3.3 Suspend/Resume and Shared Memory

On POWER7 systems, Shared Memory partitions can also be Suspend capable partitions. Shared Memory partitions require PowerVM Enterprise Edition. On PowerVM Enterprise Edition, the Shared Memory Pool interface is used to manage paging devices in the pool. Additionally, the Reserved Storage pool interface can be used to manage paging spaces in the pool.

Shared Memory partitions that are also suspend capable have only one single pool. The same paging space devices that are used to save suspension data are also used for Shared Memory.

Because there is only one paging space device pool for both Shared Memory and Suspend/Resume, Shared Memory partitions reuse the paging space devices at suspend operation to store data.

Pools: There is only one single paging space Device Pool for both Shared Memory Pool and Reserved Storage Device Pool. In spite of having different purposes, both pools use the same paging space devices.

Interactions between two pools are as follows:

- ▶ When Shared Memory Pool is created, Reserved Storage pool gets created,
- ▶ When Shared Memory Pool is deleted, Reserved Storage pool is *not* automatically deleted,
- ▶ When Reserved Storage pool is created, Shared memory pool is not automatically visible to the user,
- ▶ When Shared memory pool already exists, Reserved Storage pool cannot be deleted as it serves as storage provider to Shared memory pool,
- ▶ A pool cannot be deleted when the devices in pool are in use by a partition,

A comparison between PowerVM SE and PowerVM EE regarding pool management interfaces is shown in Figure 11-22.

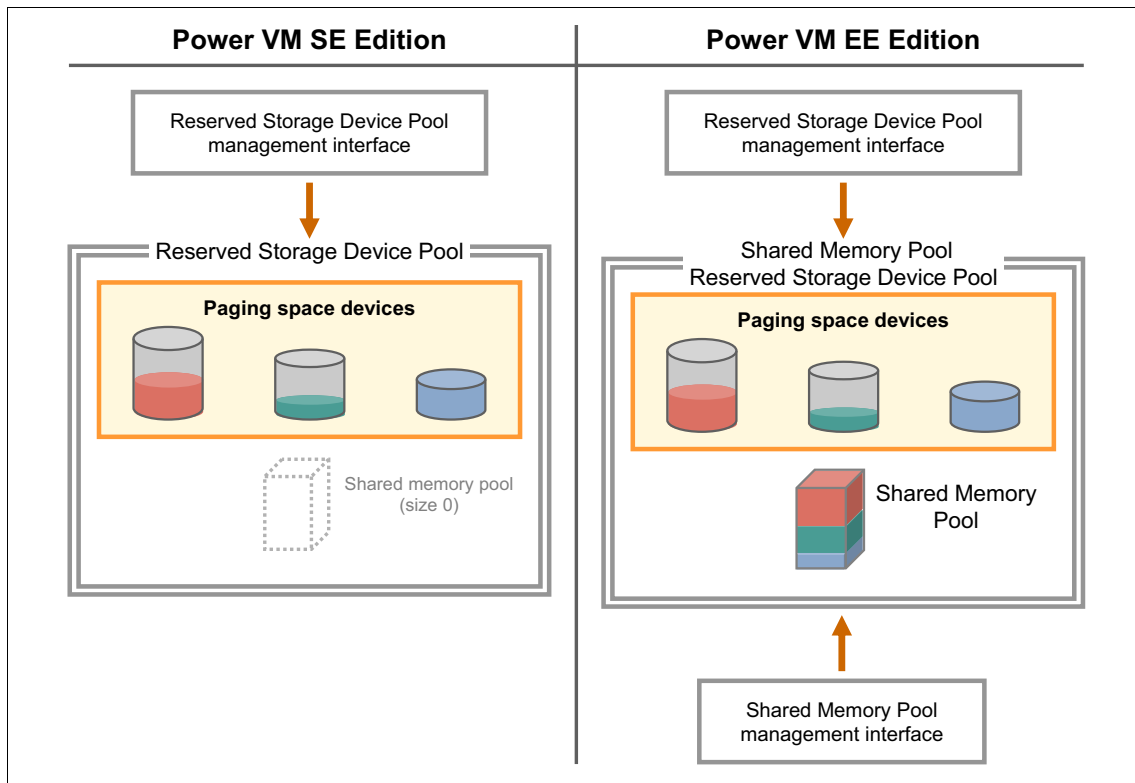


Figure 11-22 Pool management interfaces

Next we illustrate how the paging space devices are allocated from the Shared Memory Pool and Reserved Storage Device Pool.

Partitions 1 and 2 use the paging space devices 1 and 2 from the pool at the SAN storage. Similarly, partition 4 uses the local storage assigned to VIOS partition 2. The suspend/resume operations using a physical VIOS storage device has slower performance than using a SAN disk.

Partition 3 is suspend capable only and therefore it does not have allocated memory in the Shared Memory Pool. In this case, paging space device 3 represented in yellow is used to store the state for partition 3 when it is suspended. On the other hand, partitions 1, 2 and 4 are Shared Memory partitions and require allocated memory space in the Shared memory pool.

See Figure 11-23 for a diagram of these concepts.

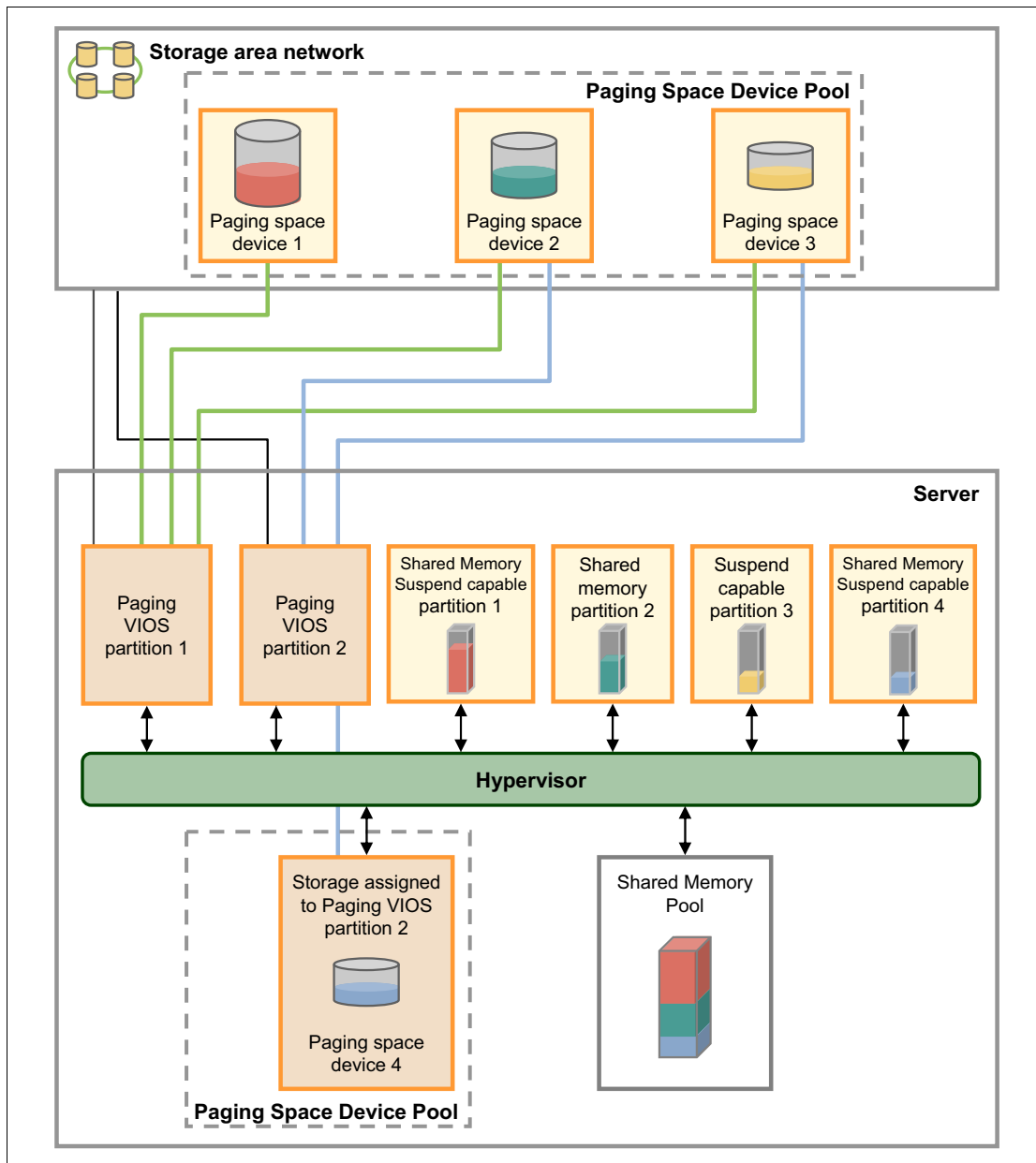


Figure 11-23 Shared Memory Pool and Reserved Storage Device Pool

Suspend

These are the HMC high level steps for suspend validation on an active partition:

1. Checks if CEC is suspend capable and if partition is active and has the Suspend setting enabled.
2. Checks for max number of concurrent suspend operations in progress.
3. Checks for presence of Reserved Storage Device Pool, and at least one active Virtual I/O Server with RMC connection.
4. Checks for the presence of restricted resources and restricted settings on the partition.
5. If partition already has a paging device, checks the size requirement. If the partition does not have a paging device, checks for availability of suitable device in the pool.
6. Checks with OS (using RMC) if it is capable of suspend and if it is ready for suspend.

These are the HMC high level steps for suspend operation on an active partition:

1. Performs validation.
2. Associates the storage device to the partition if not already associated.
3. Initiates the suspend process.
4. Keeps note of progress (at both Hypervisor and in HMC) based on Hypervisor async messages. All the HMC data transfer happens by Hypervisor through Virtual I/O Server to the storage device using VASI channel.
5. Displays the progress information to the user:
 - a. GUI: Progress bar with % complete.
 - b. CLI: Total and remaining MB with **lssyscfg** command.
6. The user has an option to stop the suspend operation. User initiated cancel of a suspend operation is accepted until the Hypervisor completes its work.
7. If the suspend operation fails, HMC auto recovers from the operation.
8. If HMC auto recover fails, the user can initiate recover explicitly.
9. After the partition is suspended, the HMC performs a cleanup operation, which involves removing virtual adapters from the Virtual I/O Servers and updating their last activated profiles.
10. After HMC cleanup, the partition power state is changed to *Suspended*.

The progress states visible in the HMC GUI are as follows:

- ▶ Starting
- ▶ Validating
- ▶ Saving HMC data
- ▶ Saving partition data
- ▶ Completing

Tip: The saving partition data step can be a long running process, depending on the memory size. It can take several minutes to complete the operation.

Resume

These are the HMC high level steps for resume validation on an active partition:

1. Checks for presence of Reserved Storage Device Pool, and at least one active Virtual I/O Server with RMC connection.
2. Reads the partition configuration data from the storage device and checks:
 - a. Partition compatibility.
 - b. If all the virtual I/O adapters can be restored.
 - c. If processor and memory types are supported and the quantity of resources for the partitions can be re-allocated.
3. If validation fails, the error information is displayed to the user, who can then decide on appropriate corrective action.

These are the HMC high level steps for resume operation on an active partition:

1. Performs validation.
2. Initiates the resume process:
 - a. HMC does the resource allocation (processor and memory), and reconfigure the partition's virtual adapters.
 - b. The Virtual I/O Server's runtime virtual adapters are updated along with its virtual adapters in its last activated profile.
3. Keeps note of progress (at both Hypervisor and in HMC) based on Hypervisor async messages.
4. Displays the progress information to the user:
 - a. GUI: Progress bar with % complete.
 - b. CLI: Total and remaining MB with `lssyscfg` command.
5. The user has an option to cancel the resume operation. User initiated cancel of the resume operation is accepted until the Hypervisor completes its work.
6. If the resume operation fails, HMC auto-recovers from the operation.
7. If HMC auto-recover fails, the user can initiate recover explicitly.

8. After the partition is resumed, the partition power state is changed to *Running*. The storage device is released after the resume operation is complete (only if not a Shared Memory partition).

The progress states visible in the HMC GUI are as follows:

- ▶ Preparing
- ▶ Validating
- ▶ Restoring partition configuration
- ▶ Reading partition data
- ▶ Completing

11.3.4 Shutdown

When a suspended partition is shut down, the HMC reconfigures all virtual adapters and hence follows the resume flow partially. This ensures subsequent activation of the partition with last activated profile succeeds.

A *force* shutdown option is available if virtual adapter reconfiguration faces an unrecoverable error. Using the *force* option might leave the partition in an inconsistent state, especially if the paging device containing the Suspended state cannot be accessed.

If you perform a *force* shutdown of a suspended partition, you might need to manually clear the paging device that was used to contain the suspended state of the partition. Otherwise the paging device might be left in a state that will prevent it from being used for future Suspend/Resume operations.

When a partition is shut down, the paging device is released if not a Shared Memory partition.

Shutdown: The normal process is to resume the partition and manually shut down. Force shutdown of a suspended partition can result in problems on the next reboot.

11.3.5 Recover

A user can issue a recover in one of the following situations:

- ▶ Suspend/Resume is taking a long time and the user ends the operation abruptly.
- ▶ The user is not able to abort Suspend/Resume successfully.
- ▶ Initiating a Suspend/Resume has resulted in an extended error indicating that the partition's state is not valid.

HMC determines the last successful step in the operation from progress data, which is stored on both HMC and Hypervisor. Based on the last successful step, HMC tries to either proceed further to continue the operation or rollback the operation. If there is no progress data available, the user has to use the force option to recover. In this case HMC recovers as much as possible. The user can recover operation using the same HMC or a different HMC.

11.3.6 Migrate

Live Partition Mobility (LPM) allows the movement of logical partitions from one server to another. LPM requires PowerVM Enterprise Edition.

A suspended logical partition can be migrated between two POWER7 technology-based systems if the destination system has enough resources to host the partition.

When a suspended partition is migrated to another CEC, the partition's profile and configuration data are moved to the destination. The partition can be resumed at a later stage on the destination CEC to which it was migrated.



Part 3

Install

This part of the publication shows how to create and install the Virtual I/O Server and client partitions. It includes information on AIX, IBM i, and Linux installations.

This part includes the following topics:

- ▶ I/O virtualization implementation
- ▶ Server virtualization implementation



I/O virtualization implementation

This chapter describes the creation and installation of a virtual I/O Server.

It covers the following topics:

- ▶ Creating a Virtual I/O Server
- ▶ Installation of Virtual I/O Server
- ▶ Defining virtual disks for client partitions

12.1 Creating a Virtual I/O Server

This section describes the steps to create a Virtual I/O Server logical partition on the HMC, install the Virtual I/O Server software and configure the Virtual I/O Server for providing virtualized devices to its client partitions.

For demonstration purposes, we will use a simple configuration consisting of a single Virtual I/O Server partition servicing virtual SCSI devices and a single network to four logical client partitions as shown in Figure 12-1.

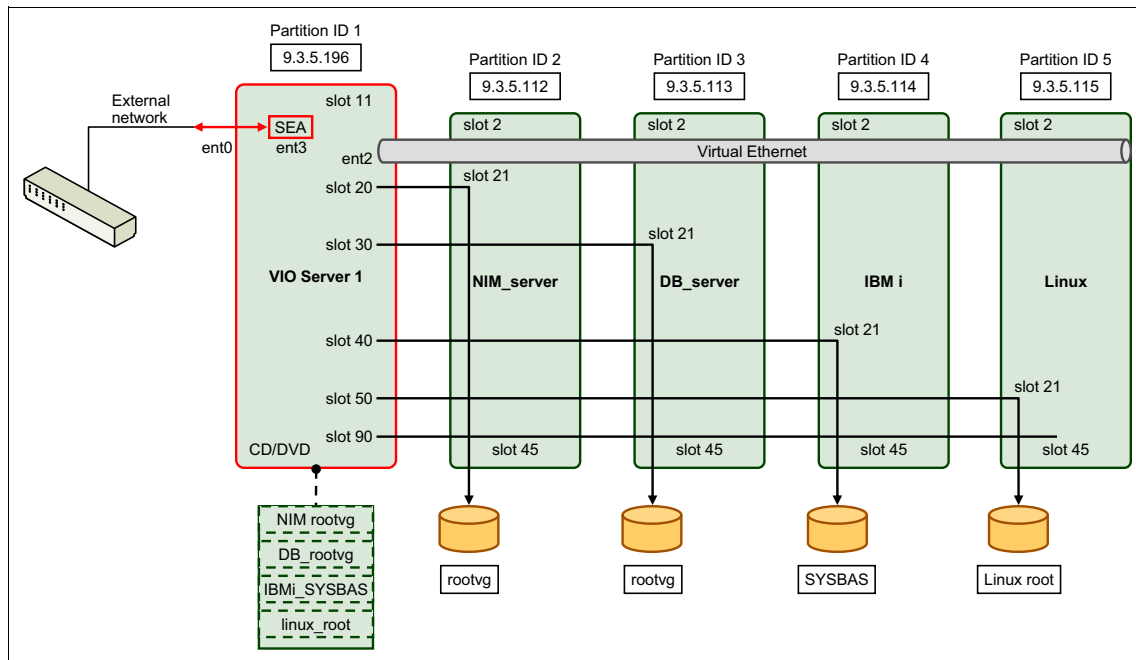


Figure 12-1 Basic Virtual I/O Server scenario

12.1.1 Creating the Virtual I/O Server partition

Experience has shown that a shared, uncapped partition is adequate in most cases to use for the Virtual I/O Server.

Tip: If you do not have access to a NIM server, you can configure NIM in a partition temporarily for easy installation of partitions.

Figure 12-2 shows the HMC with two attached managed systems. For our basic Virtual I/O Server configuration setup example, we use the managed system named p570_170. This is a chosen name based on part of the serial number for easy identification on the HMC.

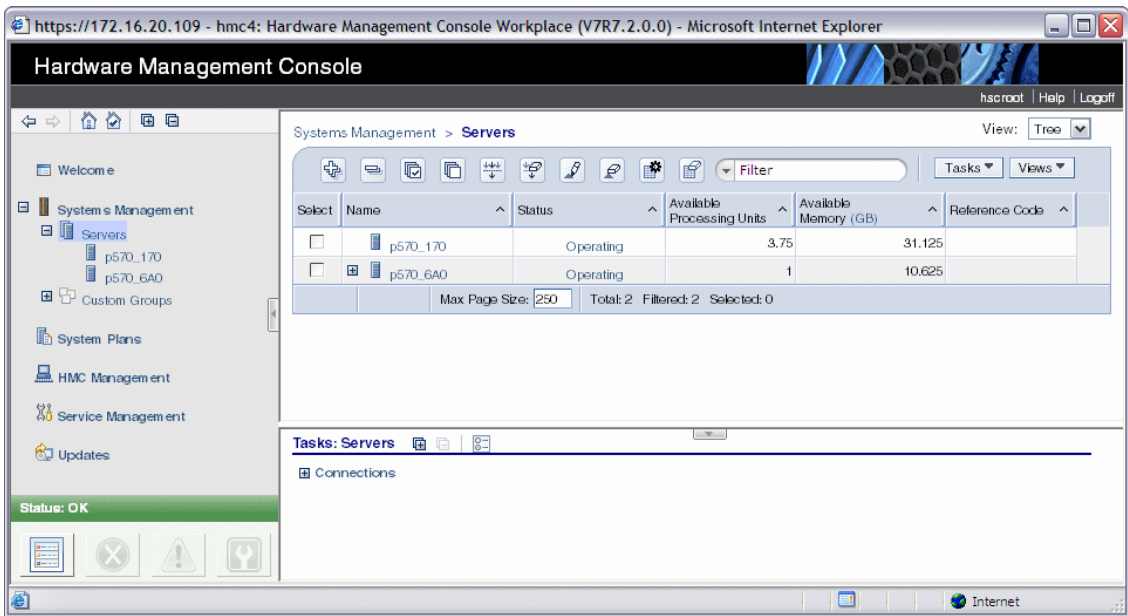


Figure 12-2 Hardware Management Console server view

User interface: The user interface of HMC V7 has changed from earlier versions. The HMC V7 Web browser-based interface is intuitive. See also the *IBM Power Systems HMC Implementation and Usage Guide*, SG24-7491 for more information.

In the following panels, we create our first Virtual I/O Server partition:

1. Select the managed system **p570_170**, then select **Configuration** → **Create Logical Partition** → **VIO Server** as shown in Figure 12-3, to start the Create Logical Partition Wizard.

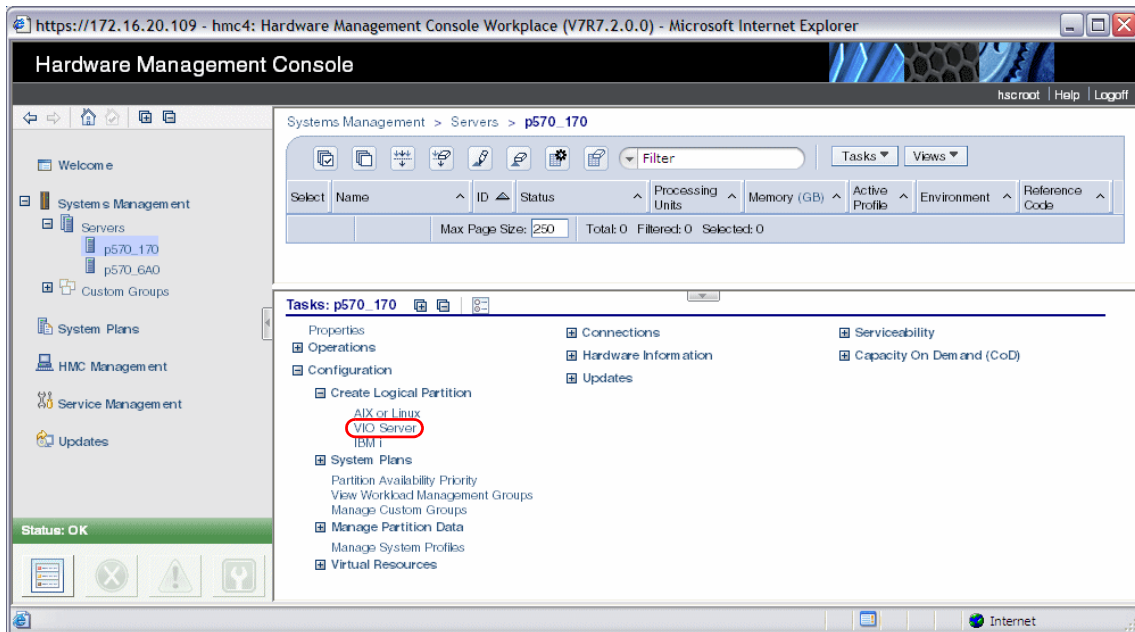


Figure 12-3 HMC Starting the Create Logical Partition wizard

2. Enter the partition name and ID as shown in Figure 12-4. – or keep the ID selected by the HMC; IDs must be unique.

Attention: Leave the **Mover service partition** box checked, if the Virtual I/O Server partition to be created must support Partition Mobility.

Click **Next** to continue.

https://172.16.20.109 - Create Lpar Wizard : p570_170 - Microsoft In...

Create Lpar Wizard : p570_170

→ **Create Partition**

- Partition Profile
- Processors
- Processing Settings
- Memory Settings
- I/O
- Virtual Adapters
- Logical Host Ethernet Adapters (LHEA)
- Optional Settings
- Profile Summary

Create Partition

This wizard helps you create a new logical partition and a default profile for it. You can use the partition properties or profile properties to make changes after you complete this wizard.

To create a partition, complete the following information:

System name : p570_170
Partition ID : 1
Partition name : VIO_Server1

Partition migration:

☒ Mover service partition

< Back Next > Finish Cancel Help

Done Internet

Figure 12-4 HMC Defining the partition ID and partition name

3. Give the partition a profile name as shown in Figure 12-5. This becomes the default profile. A partition can have several profiles and you can change which profile is default. Click **Next** to continue.



Figure 12-5 HMC Naming the partition profile

Attention: If the check box **Use all resources in the system** is checked, the logical partition being defined will get all the resources in the managed system and the partition will behave like a single server.

4. Select whether processors are to be part of a shared pool or dedicated for this partition. If shared is selected, it means that this will be a micro-partition. See Figure 12-6. Click **Next** to continue.

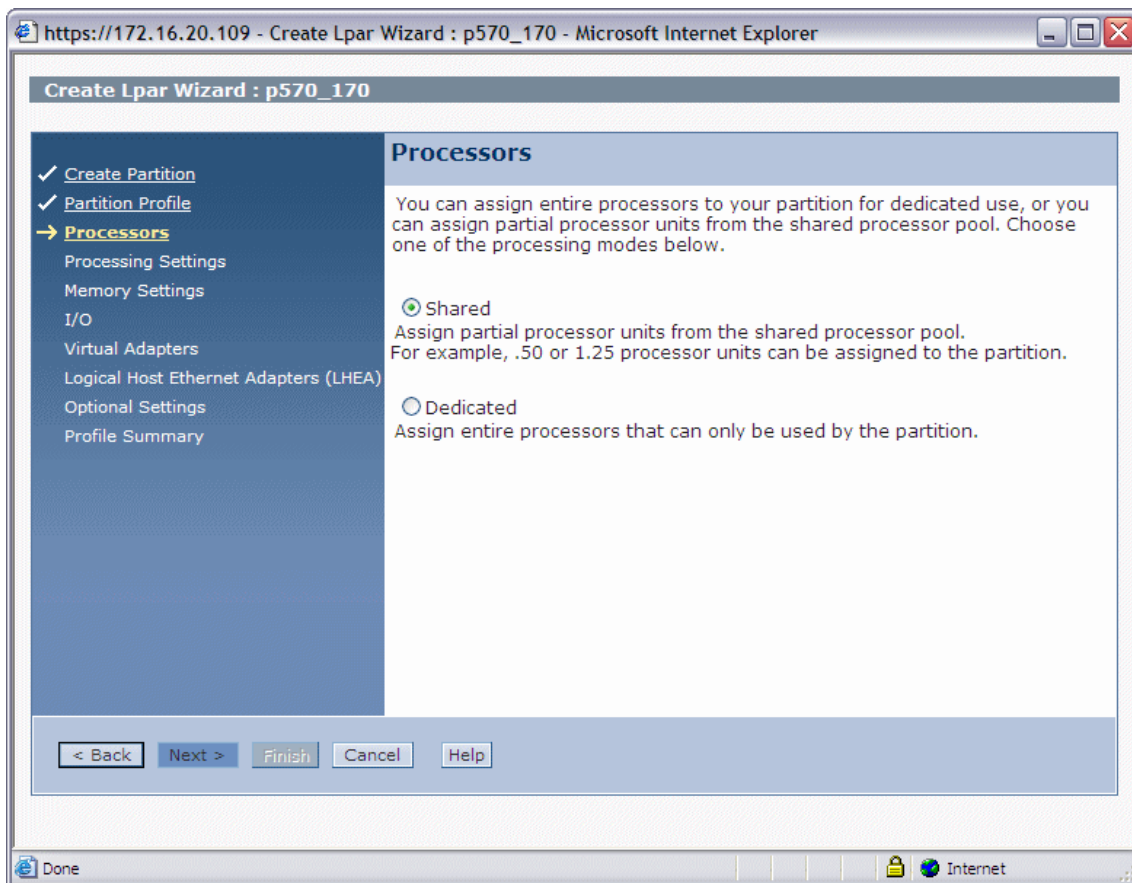


Figure 12-6 HMC Select whether processors are to be shared or dedicated

5. Figure 12-7 shows the Processing Settings for micro-partitions. We increased the default weight of 128 to 191 because this is a Virtual I/O Server partition that must have priority.

https://172.16.20.109 - Create Lpar Wizard : p570_170 - Microsoft Internet Explorer

Create Lpar Wizard : p570_170

- ✓ Create Partition
- ✓ Partition Profile
- ✓ Processors
- **Processing Settings**
- Memory Settings
- I/O
- Virtual Adapters
- Logical Host Ethernet Adapters (LHEA)
- Optional Settings
- Profile Summary

Processing Settings

Specify the desired, minimum, and maximum processing settings in the fields below.

Total usable processing units: 4.00

Minimum processing units: * 0.25

Desired processing units: * 1

Maximum processing units: * 2

Shared processor pool: DefaultPool (0)

Virtual processors

Minimum processing units required for each virtual processor: 0.10

Minimum virtual processors: * 1

Desired virtual processors: * 1

Maximum virtual processors: * 2

☒ Uncapped

Weight : 191.0

< Back Next > Finish Cancel Help

Done Internet

Figure 12-7 HMC Virtual I/O Server processor settings for a micro-partition

If you want to exploit the Multiple Shared Processor Pools capabilities and you have defined shared processor pools, you can specify here which pool this partition must belong to. See 14.1, “Configuring Multiple Shared-Processor Pools” on page 388.

Click **Next** to continue.

Rules: The following rules apply to the processor settings:

- ▶ The system will try to allocate the desired values.
- ▶ The partition will not start if the managed system cannot provide the minimum amount of processing units.
- ▶ You cannot dynamically increase the amount of processing units to more than the defined maximum. If you want more processing units, the partition needs to be stopped and reactivated in order to read the updated profile (not just rebooted).
- ▶ The maximum number of processing units cannot exceed the total Managed System processing units.

Reference: See 1.4.3, “Micro-partitioning” on page 18 for more information about processing units, capped and uncapped mode, and virtual processors.

6. Choose the memory settings, as shown in Figure 12-8.

https://172.16.20.109 - Create Lpar Wizard : p570_170 - Microsoft Internet Explorer

Create Lpar Wizard : p570_170

- ✓ [Create Partition](#)
- ✓ [Partition Profile](#)
- ✓ [Processors](#)
- ✓ [Processing Settings](#)
- [Memory Settings](#)
- [I/O](#)
- [Virtual Adapters](#)
- [Logical Host Ethernet Adapters \(LHEA\)](#)
- [Optional Settings](#)
- [Profile Summary](#)

Memory Settings

Physical Memory

Installed Memory 32768

Current memory available for Partition usage (MB) 31872

Minimum Memory 0 GB 768 MB

Desired Memory 4 GB 0 MB

Maximum Memory 8 GB 0 MB

< Back Next > Finish Cancel Help

Internet

Figure 12-8 HMC Virtual I/O Server memory settings

Rules: The following rules apply to the system shown in Figure 12-8 on page 320:

- ▶ The system will try to allocate the desired values.
- ▶ If the managed system is not able to provide the minimum amount of memory, the partition will not start.
- ▶ You cannot dynamically increase the amount of memory in a partition to more than the defined maximum. If you want more memory than the maximum, the partition needs to be stopped and the profile updated and then restarted.
- ▶ The ratio between minimum amount of memory and maximum cannot be more than 1/64.

Click **Next** to continue.

7. Select the physical I/O adapters for the partition as shown in Figure 12-9. Required means that the partition will not be able to start unless these are available to this partition. Desired means that the partition can start also without these adapters.

Click **Add as required**.

We are creating a Virtual I/O Server partition, which in our case requires a Fibre Channel adapter to attach SAN disks for the client partitions. It also requires an Ethernet adapter for Shared Ethernet adapter bridging to external networks. The SAS adapter is attached to the internal disks and the DVD-RAM on this POWER6 system.

The latest versions of the Virtual I/O Server require a minimum of 30 GB of disk space to store the installed contents of the installation media.

Click **Next** to continue.

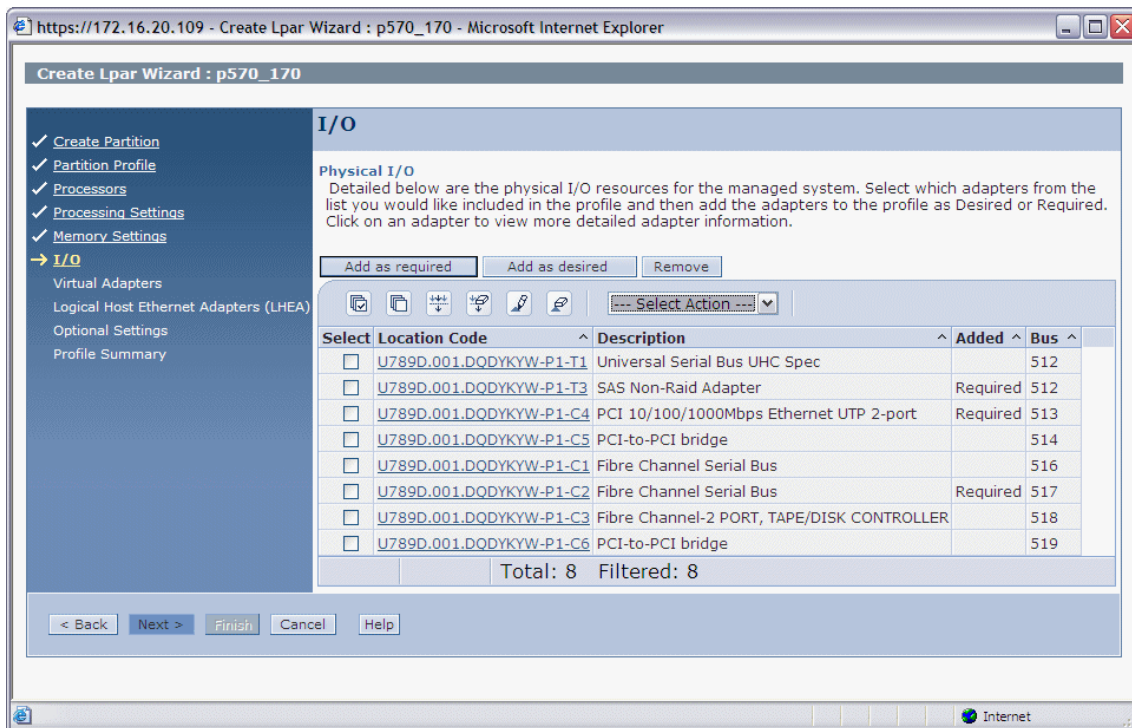


Figure 12-9 HMC Virtual I/O Server physical I/O selection for the partition

Adapters:

- ▶ The adapters for the Virtual I/O Server are set to *required* because they are needed to provide I/O to the client partitions.
- ▶ A required adapter cannot be moved in a dynamic LPAR operation.
- ▶ To change the setting from *required* to *desired* for an adapter, you have to change the profile, stop, and restart the partition.

Considerations:

- ▶ *Do not* set the adapter (if separate adapter) that holds the DVD to *required*, as it might be moved in a dynamic LPAR operation later.
- ▶ The installed Ethernet adapter in this system is a 2-port adapter. Both ports are owned by the same partition. In general, all devices attached to an adapter are owned by the partition that holds the adapter.
- ▶ Virtualization will usually reduce the required number of physical adapters and cables.
- ▶ If possible, use hot-plug Ethernet adapters for the Virtual I/O Server for increased serviceability.

8. Create virtual Ethernet and virtual SCSI adapters. The start menu is shown in Figure 12-10. We increased the maximum number of virtual adapters to 100 to allow for a flexible numbering scheme.

Important: The default serial adapters are required for console login from the HMC. Do not change these.

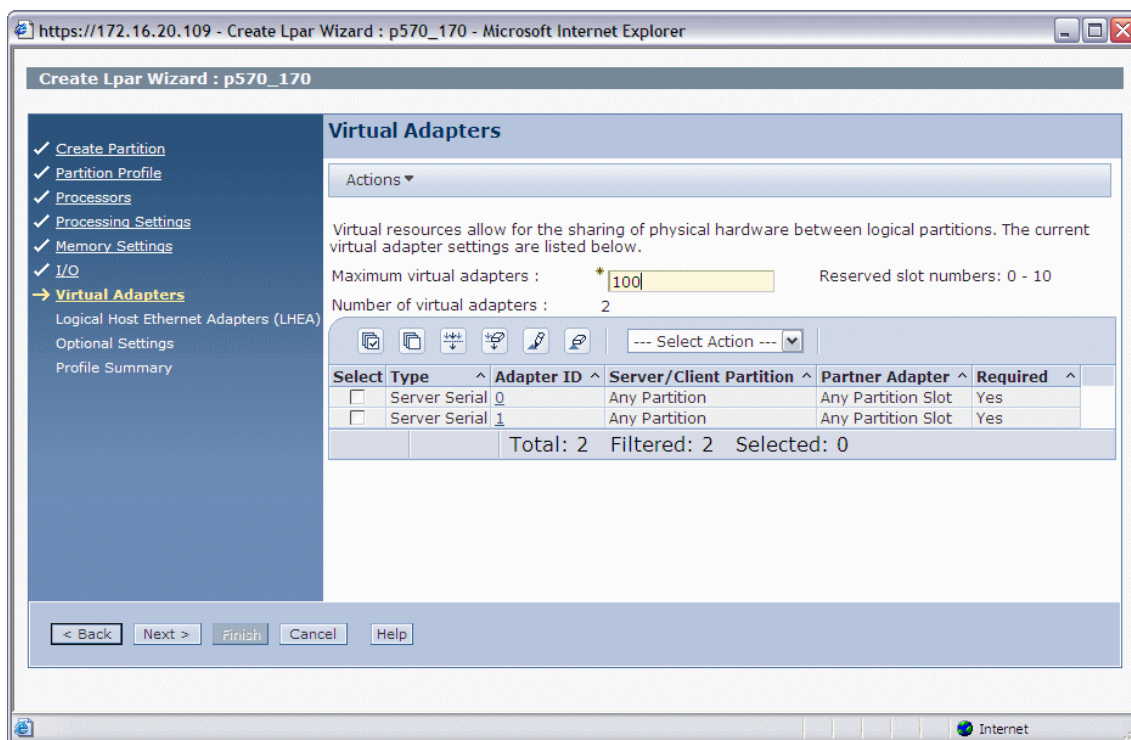


Figure 12-10 HMC start menu for creating virtual adapters

Attention: The maximum number of adapters must not be set above 1024.

- Click **Actions** and select **Create Virtual Adapter** → **Ethernet Adapter** as shown in Figure 12-11 to open the Create Virtual Ethernet Adapter window.

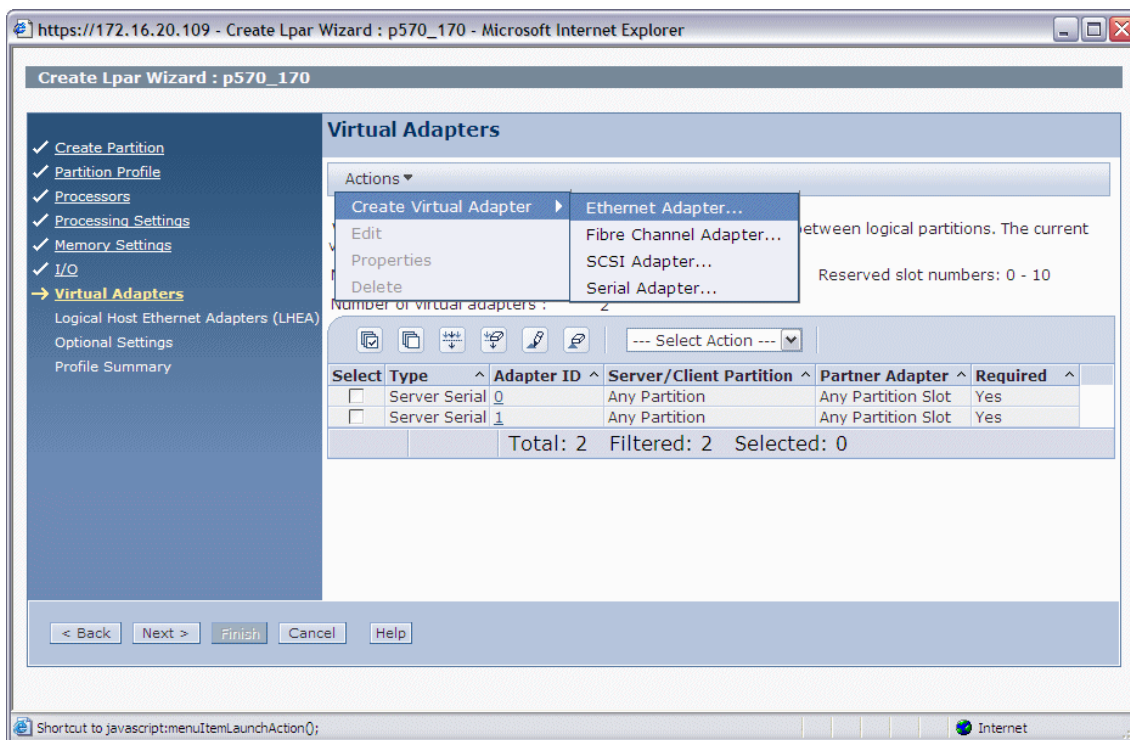


Figure 12-11 HMC Selecting to create a virtual Ethernet adapter

10. A Virtual Ethernet adapter is a logical adapter that emulates the function of a physical I/O adapter in a logical partition. Virtual Ethernet adapters enable communication to other logical partitions within the managed system without using physical hardware and cabling. A Virtual I/O Server is only required for communication to an external network. Input the Adapter ID (in this case 11) and a VLAN ID (in this case 1) as shown in Figure 12-12. IEEE 802.1q is not needed here because VLAN tagging is not used.

Select the **Access External network** (in a later HMC version, it is **Use this adapter for Ethernet bridging**) check box to use this adapter as a gateway between an internal and an external network. This virtual Ethernet will be configured as part of a Shared Ethernet Adapter. You can select the **IEEE 802.1Q compatible adapter** check box if you want to add additional virtual LAN IDs.

Click **OK** when finished.

You can create more adapters if you need more networks.

Reference: For more information about trunk priority, see 16.3.2, “SEA failover” on page 591.

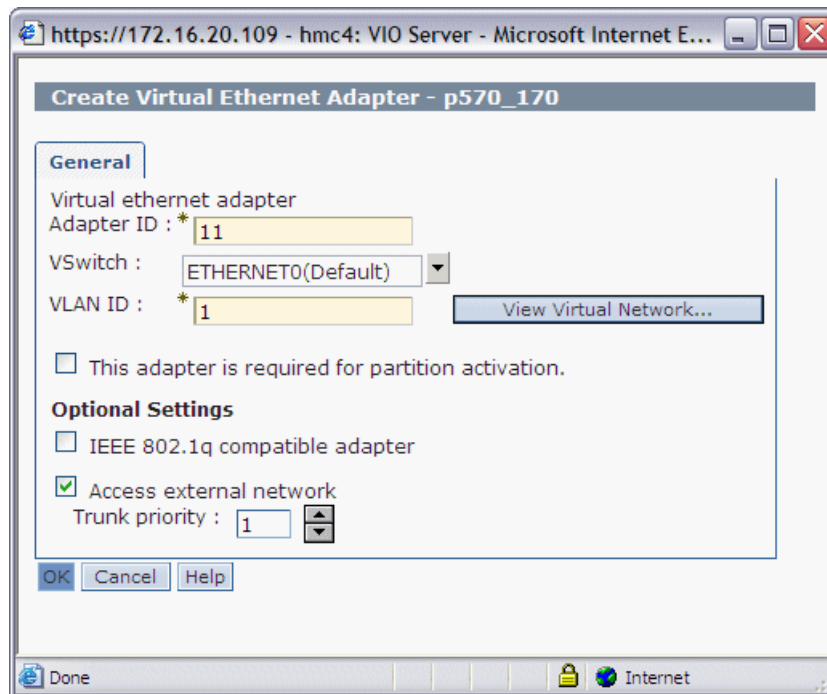


Figure 12-12 HMC Creating the virtual Ethernet adapter

Considerations:

- ▶ Adapter ID and slot ID are used interchangeably.
- ▶ Selecting the **Access External Networks** check box makes sense only for a Virtual I/O Server partition. Do not select this flag when configuring the client partition's virtual Ethernet adapters.

Tip: You can create an additional virtual Ethernet adapter for the Virtual I/O Server if you prefer to configure the IP address on a separate adapter instead of the SEA.

11. In the Virtual Adapters dialog, click **Actions** and select **Create Virtual Adapter** → **SCSI Adapter** to open the Create Virtual SCSI Adapter window shown in Figure 12-13.

Create a server adapter to be used by the virtual optical device (CD or DVD). Adapter number 90 is used here but you can use any unique number that fits your configuration. Note that this adapter is set to **Any client partition can connect**. Click **OK** when finished. This dedicated adapter for the virtual optical device helps to make things easier from a system management point of view.

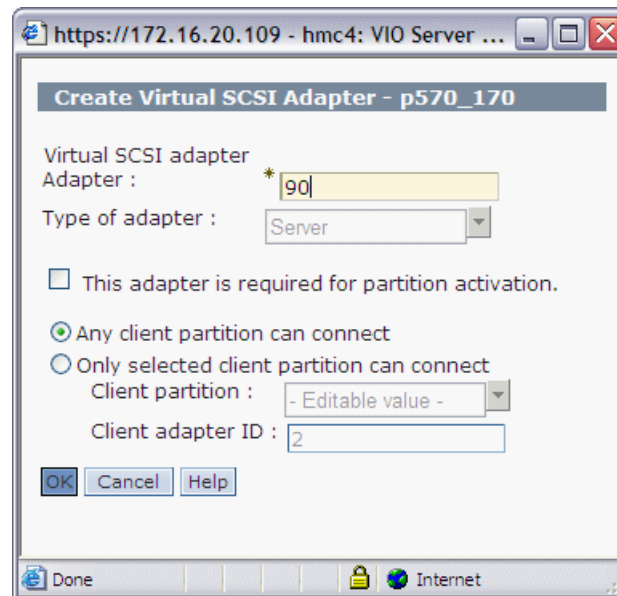


Figure 12-13 HMC Creating the virtual SCSI server adapter for the DVD

Adapters: For virtual server adapters, it is not necessary to check the box. This adapter is required for partition activation.

12. Click **Actions** and select **Create Virtual Adapter** → **SCSI Adapter** again to open the Create Virtual SCSI Adapter window shown in Figure 12-14 to create additional virtual SCSI adapters for client partition disks. We will create 4 client partitions later, so we need 4 server adapters with slot numbers 20, 30, 40, 50. Client slot is set to 21 for all clients.

At this stage, the clients are not known to the HMC. If you create the SCSI server adapters now, you will have to specify the partition ID in the Client Adapter field or specify that **Any client partition can connect**, which means you will have to change this after you have created the client partitions. We plan to have partition ID of 2, 3, 4, and 5 (the Virtual I/O Server is 1). The HMC will use the partition names when the client partitions have been created. Click **OK** when finished.

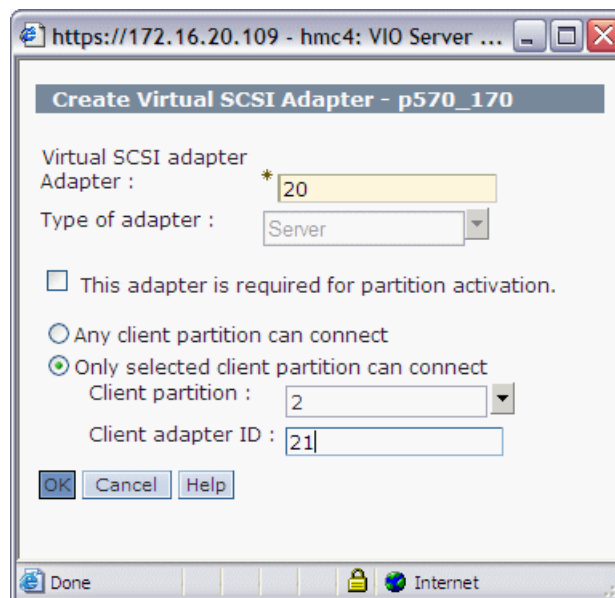


Figure 12-14 HMC Virtual SCSI server adapter for the NIM server

13. Repeat step 12 to create the SCSI server adapters for the rest of the client partitions according to Figure 12-1 on page 312. Figure 12-15 shows the list of created virtual adapters and their slot numbers. Click **Next** to continue.

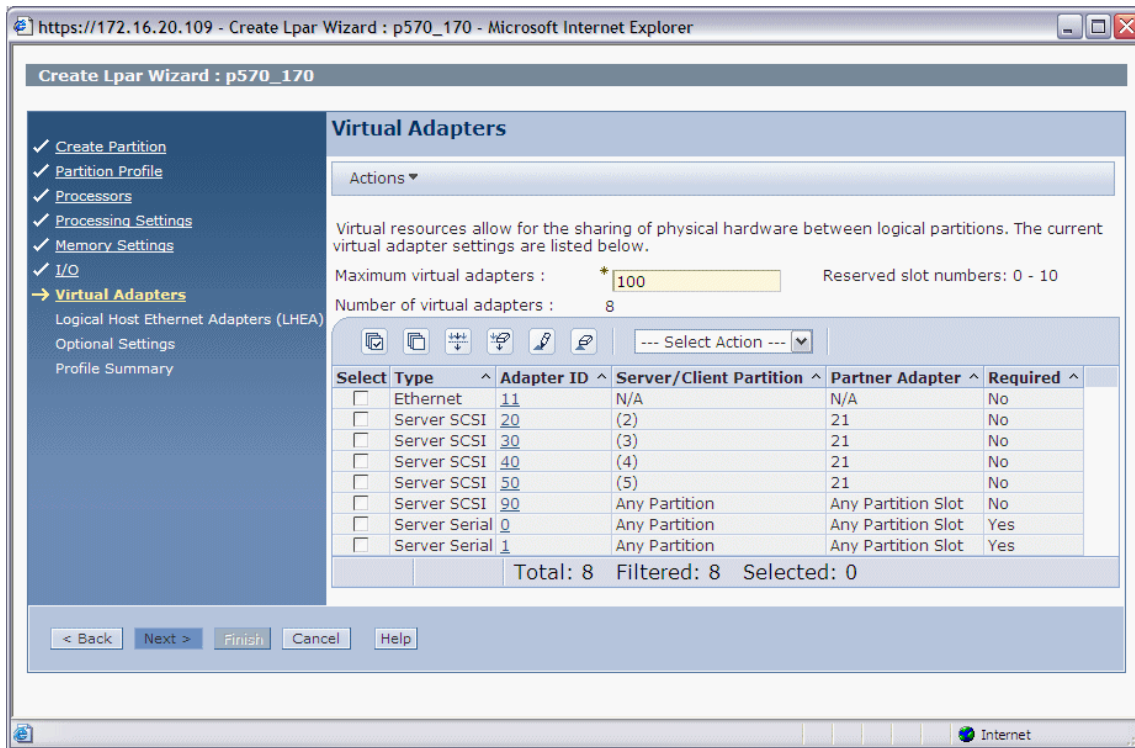


Figure 12-15 HMC List of created virtual adapters

14. The Host Ethernet Adapter, HEA, is an offering on POWER6 or later systems. It replaces the integrated Ethernet ports on selected systems and can provide logical Ethernet ports directly to the client partitions without using the Virtual I/O Server (Figure 12-16). Click **Next** to continue.

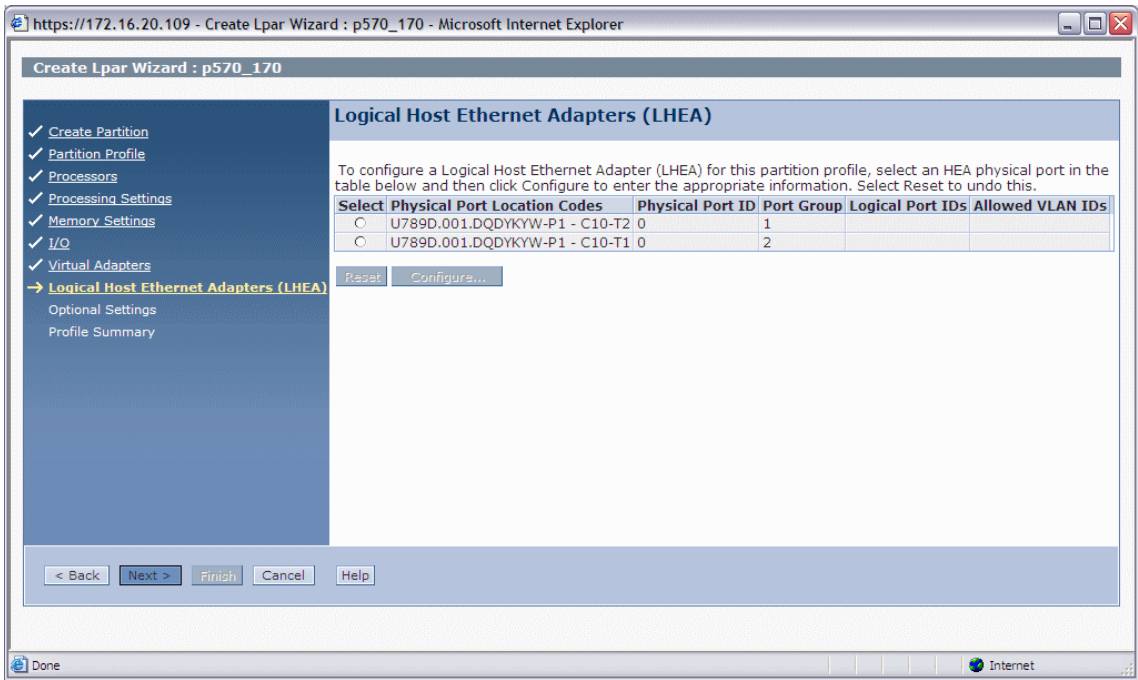


Figure 12-16 HMC Menu for creating Logical Host Ethernet Adapters

Note: The Host Ethernet Adapter will be sunset.

15. In the Optional Settings dialog for the partition, click **Next** to continue (Figure 12-17). The partition will boot in normal mode by default. “Enable connection monitoring” will alert any drop in connection to the HMC. Automatic start with managed system means that the partition will start automatically when the system is powered on with the “Partition auto start” option (selected at power-on). “Enable redundant error path reporting” allows for call-home error messages to be sent through the private network in case of open network failure.

Important: “Enable redundant error path reporting” must *not* be set for partitions that will be moved using Partition Mobility.

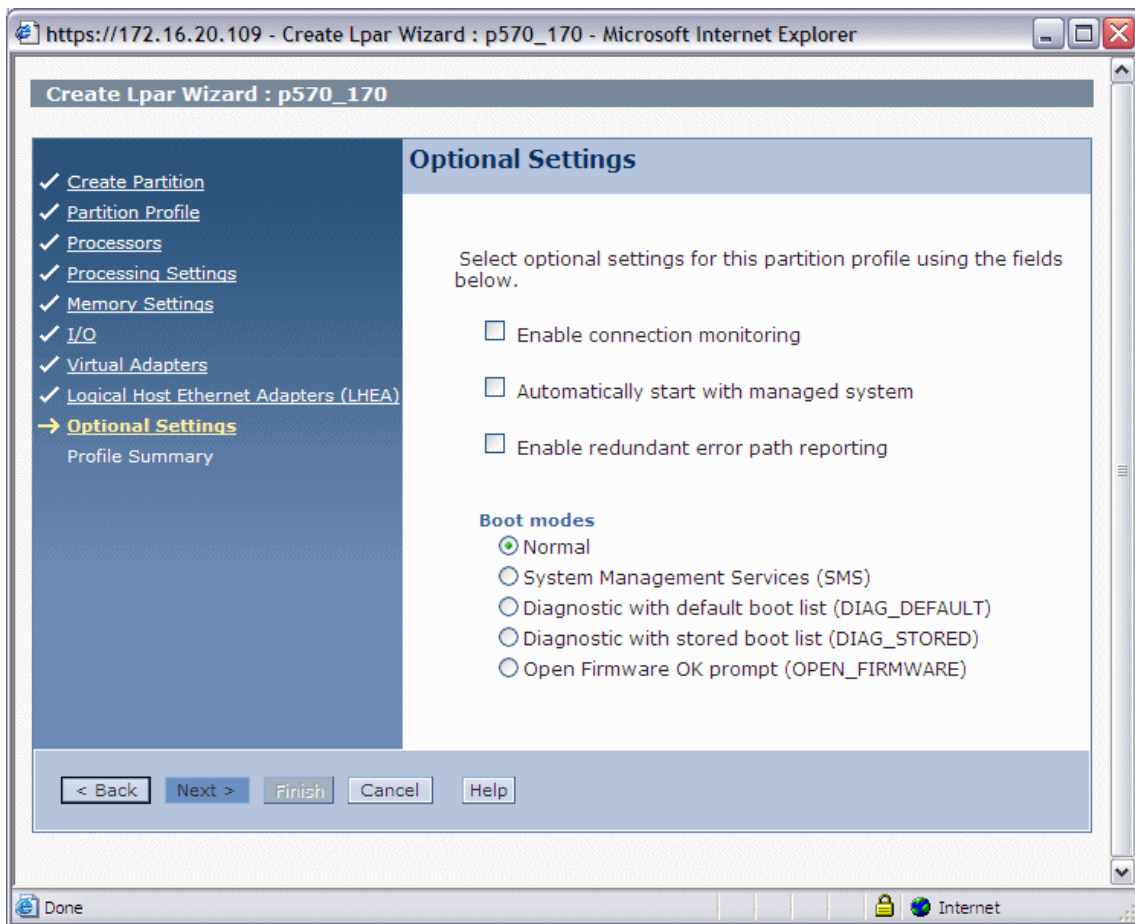


Figure 12-17 HMC Menu Optional Settings

16. The Profile Summary menu shows details about the partition configuration (Figure 12-18). You can check details about the I/O devices by clicking **Details**, or click **Back** to go back and modify any of the previous settings. Click **Finish** to complete the Virtual I/O Server partition creation.

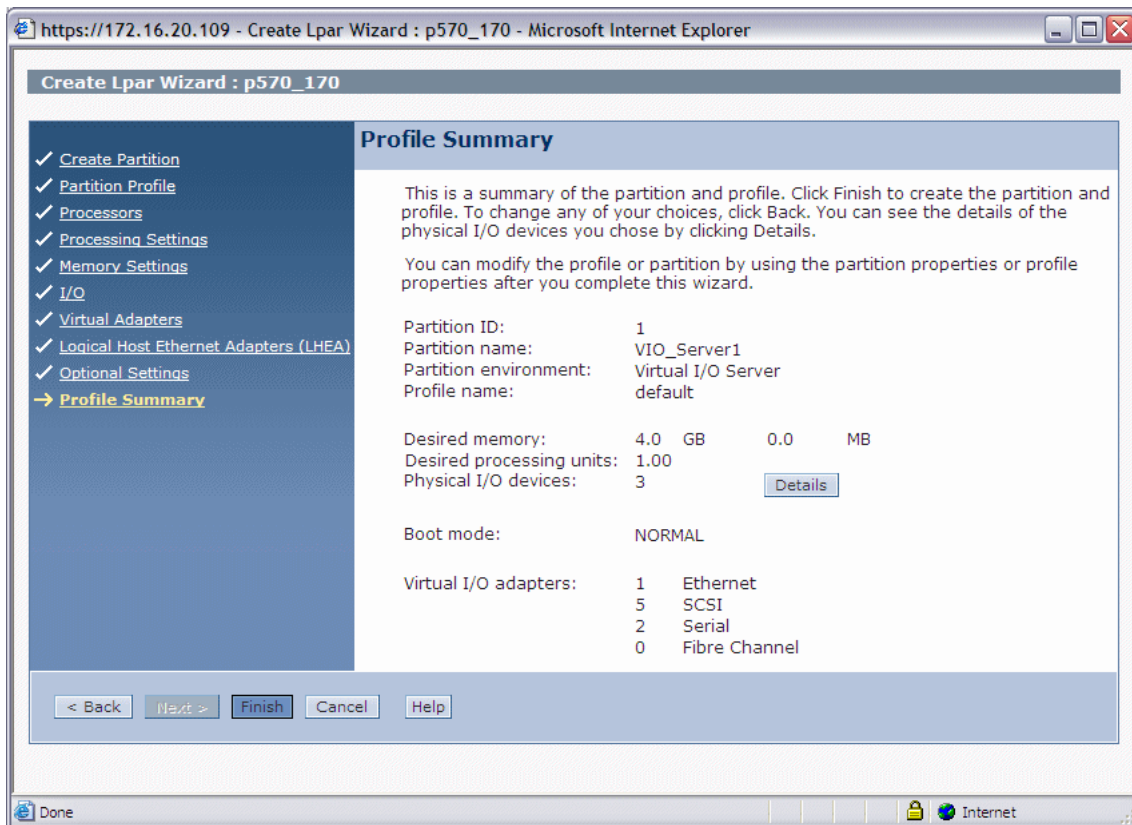


Figure 12-18 HMC Menu Profile Summary

Figure 12-19 shows the created Virtual I/O Server partition on the HMC.

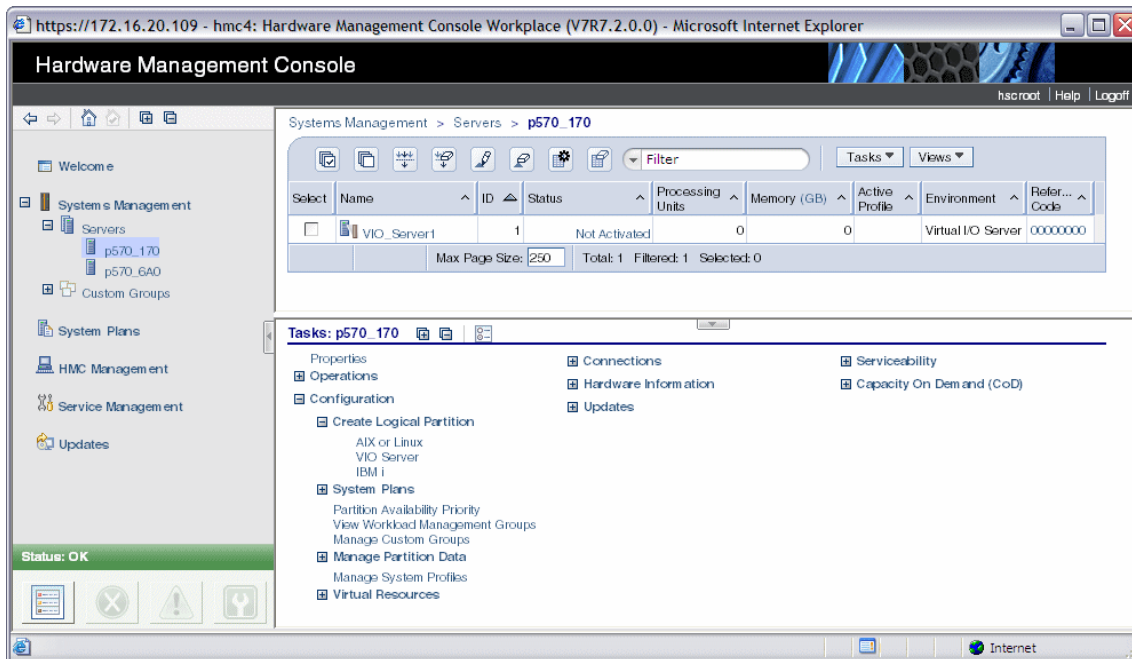


Figure 12-19 HMC The created partition VIO_Server1

12.2 Installation of Virtual I/O Server

Installation of the Virtual I/O Server partition is performed from DVD-ROM installation media that is provided to clients that order the PowerVM feature. The Virtual I/O Server software is only supported in Virtual I/O Server partitions.

The Virtual I/O Server DVD-ROM installation media can be installed in the following ways:

- Media (assigning the DVD-ROM drive to the partition and booting from the media). Detailed steps can be found in "Installing Virtual I/O Server using Optical device" on page 337.

- The HMC (inserting the media in the DVD-ROM drive on the HMC and using the **installios** command, or installing from a media image copied to the HMC). If you just enter **installios** without any flags, a wizard will be invoked and then you will be prompted to interactively enter the information contained in the flags. The default is to use the optical drive on the HMC for the Virtual I/O Server installation media, but you can also specify a remote file system instead. Detailed steps can be found “Installing the Virtual I/O Server image using installios on HMC” on page 342.

For details on how to install the Virtual I/O Server from the HMC, see the *IBM Systems Hardware Information Center* at this website:

http://publib.boulder.ibm.com/infocenter/powersys/v3r1m5/index.jsp?topic=/p7hb1/iphb1_vios_configuring_installhmc.htm

Requirement: A network adapter with connection to the HMC network is required for the Virtual I/O Server installation through the HMC.

- Using the DVD-ROM media together with the NIM server and executing the **smitty installios** command (the secure shell needs to be working between NIM and HMC).
- NIM by copying the DVD-ROM media to the NIM server and generating NIM resources for installation.

Example 12-1 shows the steps to copy the DVD mksysb image of the Virtual I/O Server on to a NIM repository.

Example 12-1 Copying the Virtual I/O Server DVD media on to a NIM server

```
# mount /cdrom
# cd /cdrom
# ls
.Version      RPMS          installp      nimol         sbin
OSLEVEL       bosinst.data  ismp          ppc           udi
README.vios   image.data   mkcd.data     root          usr
# cd usr/sys/inst.images
# ls -l
total 3429200
-rw-r--r--  1 root    system  1755750400 Jun 06 2007  mksysb_image
# cp mksysb_image /nim/images/vios1.mksysb
# cp /cdrom/bosinst.data /nim/resources
```

12.2.1 Changes on VIOS 2.2.1.0

Creating a NIM mksysb for VIOS 2.2.1.0 from the DVDs (there is a THIRD mksysb image on the second DVD of the distribution) is useful for NIM installations. Appending the third image to the result of the concatenation of the two mksysb images from the first DVD yielded a successful mksysb image.

The following steps show the same process:

1. Mount /cdrom (Mount VIOS 2.2.1.0 DVD 1 of 2)
2. `cd /cdrom/usr/sys/inst.images`
3. `cat mksysb_image mksysb_image2 > /nim/images/vios1.mksysb`
4. Unmount first DVD
5. Mount /cdrom (Mount VIOS 2.2.1.0 DVD 2 of 2)
6. `cd /cdrom/usr/sys/inst.images`
7. `cat mksysb_image mksysb_image2 >> /nim/images/vios1.mksysb`

Then proceed with NIM resource creation.

Important:

- Copy the `bosinst.data` file from the Virtual I/O Server DVD to the NIM repository and define it as a NIM resource. Specify this resource when you set up your NIM installation. The `bosinst.data` script will perform the SSH. For IVM, the DVD/CD drive is automatically virtualized from the Virtual I/O Server to be available to partitions.
- The `installios` command is not applicable for IVM-managed systems. Installation is from the Virtual I/O Server mksysb image.

The `installios` command is also available in AIX both for the NIM server and any NIM client. If you run the `installios` command on a NIM client, you are prompted for the location of the `bos.sysmgt.nim.master` fileset. The NIM client is then configured as a NIM master. Use the following link and search for `installios` for additional information:

<http://publib.boulder.ibm.com/infocenter/pseries/v6r1/index.jsp>

Tip: If you plan on using two Virtual I/O Servers (described in 10.1.2, “Redundancy considerations” on page 175), you can install the first server, apply updates, multipath drivers, and customization; then make a NIM backup and use this customized image for installing the second Virtual I/O Server.

Considerations: The architecture of POWER processor-based systems can allow swapping an optical media device used for installs between partitions by reassigning the Other SCSI Controller. The disks on these systems can be on their own SCSI Controller. On several systems, the media devices and internal storage can be part of a single “SAS Controller” group, or simply share a controller, preventing you from separating the install device and a subset of the internal disks.

With this in mind, there are two approaches to install a second VIOS.

- ▶ NIM install of the second VIOS from an AIX system using the **installios** command.
- ▶ Assign the install media to the first VIOS and install on the SAS disks that do not have the optical device as part of the group. Then unassign the installed disks and install on the disks that are part of the group.

VIOS: The VIOS has a minimum disk storage capacity requirement of 30 GB.

Installation differences on VIOS Version 2.2.1.0 or later

The VIOS software is distributed on two DVDs. When you boot from DVD 1, you are prompted to insert DVD 2.

If you want to install another language fileset after the initial installation is complete, insert the second DVD into the DVD drive and refer to the CLI **chlang** command.

Beginning with VIOS Version 2.2.1.0, the media no longer ships on a single DVD disc. Therefore, installing using the **installios** or **OS_install** commands now prompts the user to switch the media. The **installios** and **OS_install** commands on the following products have been updated to reflect this change and are now required to install VIOS Version 2.2.1.0 and later:

- ▶ HMC Version 7 Release 7.4.0, or later.
- ▶ AIX Version 6.1 Technology Level 7, or later.
- ▶ AIX Version 7.1 Technology Level 1, or later.

12.2.2 Installing Virtual I/O Server using Optical device

The following steps show the installation using the optical install device:

1. Place the Virtual I/O Server DVD install media in the drive of the Power Systems server.
2. Activate the VIO_Server1 partition by selecting the partition and clicking **Operations** → **Activate** → **Profile**, as shown in Figure 12-20.

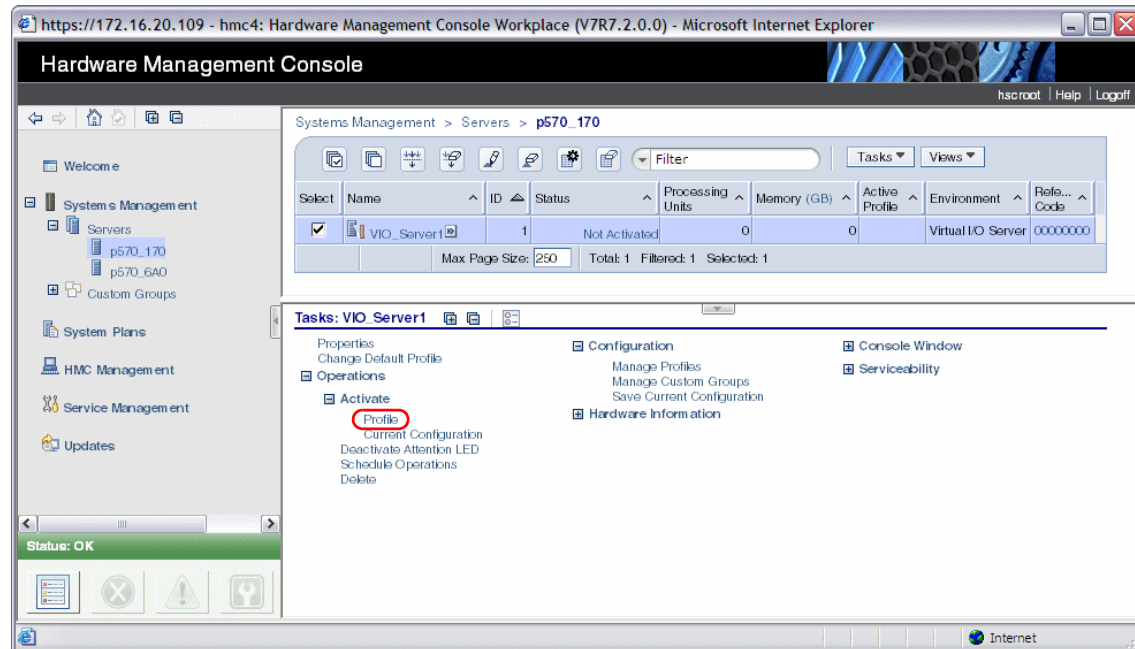


Figure 12-20 HMC Activating a partition

3. Select the default profile and then check the **Open a terminal window or console session** check box, as shown in Figure 12-21, and then click **Advanced**.

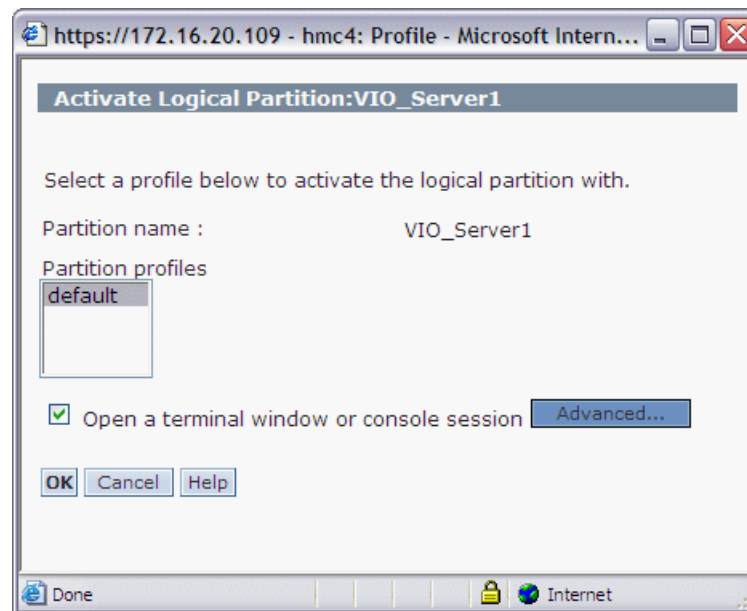


Figure 12-21 HMC Activate Logical Partition submenu

4. Under the Boot Mode drop-down list, choose **SMS**, as shown in Figure 12-22, and then click **OK**, then back in the Activate Logical partition window, click **OK** as well.

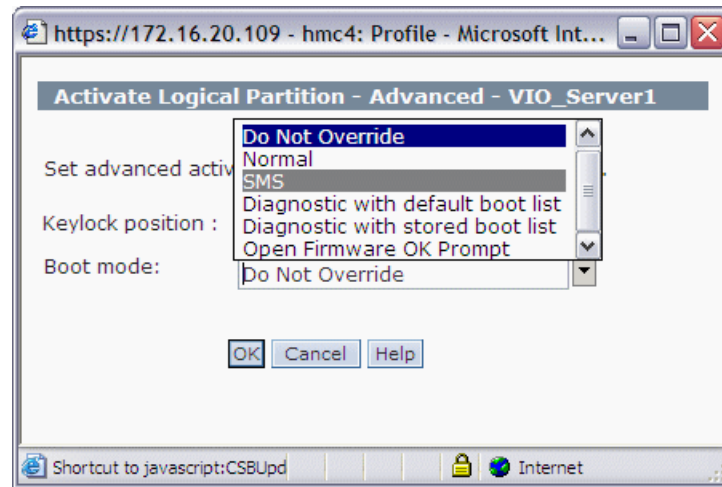


Figure 12-22 HMC Selecting the SMS menu for startup

5. Figure 12-23 shows the SMS menu after booting the partition in SMS mode.

```
Version EM350_085
SMS 1.7 (c) Copyright IBM Corp. 2000,2008 All rights reserved.
-----
--
Main Menu
1.  Select Language
2.  Setup Remote IPL (Initial Program Load)
3.  Change SCSI Settings
4.  Select Console
5.  Select Boot Options

-----

Navigation Keys:

X = eXit System Management

Services

-----

Type menu item number and press Enter or select Navigation key:
```

Figure 12-23 The SMS startup menu

6. Follow these steps to continue and boot the Virtual I/O Server partition. The process is similar to installing AIX:
 - a. Choose **5. Select Boot Options** and then press Enter.
 - b. Choose **1. Select Install/Boot Device** and then press Enter.
 - c. Choose **7. List all Devices**, look for the CD-ROM (press n to get to next page if required), enter the number for the CD_ROM and then press Enter.

Tip: It is useful to list all devices to check that the required devices are available. Only adapters will be visible at this point unless underlaying disks have a boot device on them.

- d. Choose **2. Normal Mode Boot** and then press Enter.
- e. Confirm your choice with selecting **1. Yes** and then press Enter.
- f. Next you are prompted to accept the terminal as console and then to select the installation language.
- g. You are then presented with the installation menu. It is best to check the settings (option 2) before proceeding with the installation. Check if the selected installation disk is correct.

See the IBM Systems Hardware Information Center for more information about the installation process:

<http://publib.boulder.ibm.com/infocenter/powersys/v3r1m5/index.jsp>

7. When the installation procedure has finished, use the padmin user name to log in. Upon initial login, you will be asked to supply the password. There is no default password.

After logging in successfully, you will be placed under the Virtual I/O Server command line interface (CLI).

Enter **a** (and press Enter) to accept the Software Maintenance Agreement terms, then type in the following command to accept the license:

```
$ license -accept
```

You are now ready to use the newly installed Virtual I/O Server software.

Updates: Before actually using the Virtual I/O Server, consider updating it to the latest Virtual I/O Server fix pack to benefit from latest enhancements and fixes (see “Updating the Virtual I/O Server using fix packs” on page 344).

12.2.3 Installing the Virtual I/O Server image using installios on HMC

You can also use the `installios` command on the HMC to install the Virtual I/O Server image, either stored on the HMC or the Virtual I/O Server install media in the HMC's DVD drive. Example 12-2 shows the dialog from the `installios` command without options. In our example we installed from the image on the install media inserted into the HMC's DVD drive. If you had previously copied this image to the HMC hard disk, you can specify the location of the copied image instead of the DVD drive.

Example 12-2 Running installios on the HMC

```
hscroot@hmc4:~> installios
```

```
The following objects of type "managed system" were found. Please
select one:
```

1. p570_170
2. p570_6A0

```
Enter a number (1-2): 2
```

```
The following objects of type "virtual I/O server partition" were
found. Please select one:
```

1. VIO_Server1

```
Enter a number: 1
```

```
The following objects of type "profile" were found. Please select one:
```

1. default

```
Enter a number: 1
```

```
Enter the source of the installation images [/dev/cdrom]: /dev/cdrom
```

```
Enter the client's intended IP address: 172.16.20.191
```

```
Enter the client's intended subnet mask: 255.255.252.0
```

```
Enter the client's gateway: 172.16.20.109
```

```
Enter the client's speed [100]: auto
```

```
Enter the client's duplex [full]: auto
```

```
Would you like to configure the client's network after the
installation [yes]/no? no
```

```
Please select an adapter you would like to use for this installation.
```

```
(WARNING: The client IP address must be reachable through this adapter!
```

1. eth0 10.1.1.109

```
2. eth1 172.16.20.109
3. eth2 10.255.255.1
4. eth3
Enter a number (1-4): 2
```

Retrieving information for available network adapters
This will take several minutes...

The following objects of type "ethernet adapters" were found. Please select one:

```
1. ent U9117.MMA.100F6A0-V1-C11-T1 a24e5655040b /vdevice/l-lan@3000000b
n/a virtual
2. ent U789D.001.DQDWWHY-P1-C10-T2 00145e5e1f20
/lhea@23c00100/ethernet@23e00000 n/a physical
3. ent U789D.001.DQDWWHY-P1-C5-T1 001125cb6f64
/pci@800000020000202/pci@1/ethernet@4 n/a physical
4. ent U789D.001.DQDWWHY-P1-C5-T2 001125cb6f65
/pci@800000020000202/pci@1/ethernet@4,1 n/a physical
5. ent U789D.001.DQDWWHY-P1-C5-T3 001125cb6f66
/pci@800000020000202/pci@1/ethernet@6 n/a physical
6. ent U789D.001.DQDWWHY-P1-C5-T4 001125cb6f67
/pci@800000020000202/pci@1/ethernet@6,1 n/a physical
```

```
Enter a number (1-6): 3
Enter a language and locale [en_US]: en_US
Here are the values you entered:
```

```
managed system = p570_6A0
virtual I/O server partition = VIO_Server1
profile = default
source = /dev/cdrom
IP address = 172.16.20.191
subnet mask = 255.255.252.0
gateway = 172.16.20.109
speed = auto
duplex = auto
configure network = no
install interface = eth1
ethernet adapters = 00:11:25:cb:6f:64
language = en_US
```

Press enter to proceed or type Ctrl-C to cancel...

Tips:

- ▶ If you answer *yes* to configure the client network and you want to use this physical adapter to be part of a Shared Ethernet Adapter (SEA), you will have to detach this interface to configure SEA.
- ▶ If you plan on using dual Virtual I/O Servers, it is practical to install and upgrade the first Virtual I/O Server to the desired level, then make a NIM backup and install the second Virtual I/O Server using NIM installation.

12.2.4 Updating the Virtual I/O Server using fix packs

Existing Virtual I/O Server installations can move to the latest level by applying the latest cumulative fix pack.

Fix packs provide a migration path for existing Virtual I/O Server installations. Applying the latest fix pack will update the Virtual I/O Server to the latest level. All fix packs are cumulative and contain all fixes from previous fix packs. The download page maintains a list of all fixes included with each fix pack.

Fix packs are typically a general update intended for all Virtual I/O Server installations. Fix packs can be applied to either HMC-managed or IVM-managed Virtual I/O Servers. All interim fixes applied must be manually removed before applying any fix pack.

To check the latest release and instructions for the installation of Virtual I/O Server, visit this website:

<http://www14.software.ibm.com/webapp/set2/sas/f/vios/home.html>

Tip: All Virtual I/O Server fix packs are cumulative and contain all fixes from previous fix packs. Applying the latest fix pack upgrades an existing Virtual I/O Server to the latest supported level.

For a reference about recent Virtual I/O Server version enhancements, see Appendix A, “Recent PowerVM enhancements” on page 673.

Note: Once the installation of Virtual I/O Server completes, you may need to set up your Virtual I/O Server, which includes setting up security, firewall, and also, depending on your requirement, setting up Kerberos and other aspects. See 12.1, “Creating a Virtual I/O Server” on page 312 in the setup part of this book for more details.

After completion of Virtual I/O Server software installation, define virtual disks for the client partitions.

12.3 Defining virtual disks for client partitions

Virtual disks can either be whole physical disks, logical volumes, or files. The physical disks can either be Power Systems internal disks or SAN attached disks. SAN disks can be used both for the Virtual I/O Server rootvg (using SAN boot) and for virtual I/O client disks.

Virtual disks can be defined using the Hardware Management Console or the Virtual I/O Server. More on this can be found in “Defining virtual disks” on page 466.

The Virtual I/O Server provides a command-line interface for creating and managing virtual disks. The following paragraph shows how virtual disks can be defined using the Virtual I/O Server.

12.3.1 Defining virtual SCSI disks

Virtual disks can either be whole physical disks, logical volumes, or files. The physical disks can either be Power Systems internal disks or SAN attached disks. SAN disks can be used both for the Virtual I/O Server rootvg (using SAN boot) and for virtual I/O client disks.

Use the following steps to build the logical volumes required to create the virtual disk for the client partition’s rootvg based on our basic scenario using the Virtual I/O Server:

- 1. Log in with the padmin user ID and run the **cfgdev** command to rebuild the list of visible devices used by the Virtual I/O Server.

The virtual SCSI server adapters are now available to the Virtual I/O Server. The name of these adapters will be vhostx, where x is a number assigned by the system.
- 2. Use the **lsdev -virtual** command to make sure that your five new virtual SCSI server adapters are available, as shown in Example 12-3.

Example 12-3 Listing virtual devices

\$ lsdev -virtual		
name	status	description
ent2	Available	Virtual I/O Ethernet Adapter (1-lan)
vasi0 (VASI)	Available	Virtual Asynchronous Services Interface
vhost0	Available	Virtual SCSI Server Adapter
vhost1	Available	Virtual SCSI Server Adapter
vhost2	Available	Virtual SCSI Server Adapter

vhost3	Available	Virtual SCSI Server Adapter
vhost4	Available	Virtual SCSI Server Adapter
vsa0	Available	LPAR Virtual Serial Adapter
ent3	Available	Shared Ethernet Adapter

3. Use the **lsmap -all** command to check slot numbers and vhost adapter numbers as shown in Example 12-4.

Example 12-4 Verifying slot numbers and vhost adapter numbers

```
$ lsmap -all
```

SVSA	Physloc	Client Partition ID

vhost0	U9117.MMA.101F170-V1-C20	0x00000000
VTD	NO VIRTUAL TARGET DEVICE FOUND	
SVSA	Physloc	Client Partition ID

vhost1	U9117.MMA.101F170-V1-C30	0x00000000
VTD	NO VIRTUAL TARGET DEVICE FOUND	
SVSA	Physloc	Client Partition ID

vhost2	U9117.MMA.101F170-V1-C40	0x00000000
VTD	NO VIRTUAL TARGET DEVICE FOUND	
SVSA	Physloc	Client Partition ID

vhost3	U9117.MMA.101F170-V1-C50	0x00000000
VTD	NO VIRTUAL TARGET DEVICE FOUND	
SVSA	Physloc	Client Partition ID

vhost4	U9117.MMA.101F170-V1-C90	0x00000000
VTD	NO VIRTUAL TARGET DEVICE FOUND	

If the devices are not available, then there was a problem defining them. You can use the **rmdev -dev vhost0 -recursive** command for each device and then reboot the Virtual I/O Server if needed. Upon reboot, the configuration manager will detect the hardware and recreate the vhost devices. Also check the profile on the HMC.

12.3.2 Using file-backed devices

For our basic scenario as shown in Figure 12-1 on page 312, we are only showing how to use logical volumes and physical disks from the Virtual I/O Server for virtual SCSI devices for the client partitions in the following sections.

If you are not as much concerned about the virtual SCSI I/O performance by introducing another I/O layer with the Virtual I/O Server's filesystem and prefer to use file-backed devices from the Virtual I/O Server for the virtual SCSI devices for the client partitions, use the **mksp** and **mkbdsp** commands as shown in Example 12-5.

Example 12-5 Creating a Virtual I/O Server file-backed device for a client partition

```
$ mksp -f rootvg_clients hdisk2
rootvg_clients

$ mksp -fb clients_fsp -sp rootvg_clients -size 80G
clients_fsp
File system created successfully.
83555644 kilobytes total disk space.
New File System size is 167772160

$ lssp
Pool                Size(mb)   Free(mb)   Alloc Size(mb)   BDs Type
rootvg              139776    94976      256              0 LVP00L
rootvg_clients      139904    57984      128              0 LVP00L
clients_fsp         81588     81587      128              0 FBP00L

$ mkbdsp -sp clients_fsp 20G -bd vdbsrv_rvg -vadapter vhost1
Creating file "vdbsrv_rvg" in storage pool "clients_fsp".
Assigning file "vdbsrv_rvg" as a backing device.
vtscsi0 Available
vdbsrv_rvg

$ lsmap -vadapter vhost1
SVSA                Physloc                Client Partition ID
-----
vhost1              U8233.E8B.061AA6P-V1-C30  0x00000003

VTD                vtscsi0
Status              Available
LUN                 0x8100000000000000
Backing device      /var/vio/storagepools/clients_fsp/vdbsrv_rvg
Physloc
Mirrored            N/A
```

12.3.3 Using logical volumes

In our basic scenario, we will create the volume group named `rootvg_clients` on `hdisk2` and partition it into logical volumes to serve as boot disks to our client partitions.

Considerations:

- ▶ For IBM i client partitions, consider mapping whole physical disks or SAN storage LUNs (hdisks) on the Virtual I/O Server to the IBM i client for performance reasons and configuration simplicity as described in “Using physical disks” on page 467.
- ▶ Raw logical volumes used as virtual devices by the Virtual I/O Server must have their own backup policy, because the **backupios** command does not back up raw logical volumes.
- ▶ If you choose to use raw logical volumes on `rootvg`, you need to re-create the virtual target device (VTD) after restoring.

Important: The Virtual I/O Server `rootvg` disks must not be used for virtual client disks (logical volumes).

1. Create a volume group `rootvg_clients` on `hdisk2` using the **mkvg** command, as shown in Example 12-6.

Example 12-6 Creating the rootvg_clients volume group

```
$ mkvg -f -vg rootvg_clients hdisk2
rootvg_clients
```

2. Define all the logical volumes that are going to be presented to the client partitions as hdisks using the **mklv** command. In our case, these logical volumes will be our `rootvg` for the client partitions (see Example 12-7).

Example 12-7 Create logical volumes

```
$ mklv -lv dbsrv_rvg rootvg_clients 10G
dbsrv_rvg
$ mklv -lv IBMi_LS rootvg_clients 20G
IBMi_LS
$ mklv -lv nimsrv_rvg rootvg_clients 10G
nimsrv_rvg
$ mklv -lv linux rootvg_clients 2G
linux
```

3. Define the SCSI mappings to create the virtual target device that associates to the logical volume you have defined in the previous step. Based on Example 12-8, we have four virtual host devices on the Virtual I/O Server. These vhost devices are the ones we are going to map to our logical volumes. Adapter vhost4 is the adapter for the virtual DVD. See 16.2.3, “Virtual optical” on page 490 for details on virtual optical devices.

Example 12-8 Create virtual device mappings

```
$ lsdev -vpd|grep vhost
vhost4 U9117.MMA.101F170-V1-C90 Virtual SCSI Server Adapter
vhost3 U9117.MMA.101F170-V1-C50 Virtual SCSI Server Adapter
vhost2 U9117.MMA.101F170-V1-C40 Virtual SCSI Server Adapter
vhost1 U9117.MMA.101F170-V1-C30 Virtual SCSI Server Adapter
vhost0 U9117.MMA.101F170-V1-C20 Virtual SCSI Server Adapter

$ mkvdev -vdev nimsrv_rvg -vadapter vhost0 -dev vnimsrv_rvg
vnimsrv_rvg Available
$ mkvdev -vdev dbsrv_rvg -vadapter vhost1 -dev vdbsrv_rvg
vdbsrv_rvg Available
$ mkvdev -vdev IBMi_LS -vadapter vhost2 -dev vIBMi_LS
vIBMi_LS Available
$ mkvdev -vdev linux -vadapter vhost3 -dev vlinux
vlinux Available
$ mkvdev -vdev cd0 -vadapter vhost4 -dev vcd
vcd Available

$ lsdev -virtual
name          status      description
ent2           Available  Virtual I/O Ethernet Adapter (1-lan)
vasi0          Available  Virtual Asynchronous Services Interface
(VASI)
vhost0         Available  Virtual SCSI Server Adapter
vhost1         Available  Virtual SCSI Server Adapter
vhost2         Available  Virtual SCSI Server Adapter
vhost3         Available  Virtual SCSI Server Adapter
vhost4         Available  Virtual SCSI Server Adapter
vsa0           Available  LPAR Virtual Serial Adapter
vIBMi_LS       Available  Virtual Target Device - Logical Volume
vcd            Available  Virtual Target Device - Optical Media
vdbsrv_rvg     Available  Virtual Target Device - Logical Volume
vlinux         Available  Virtual Target Device - Logical Volume
vnimsrv_rvg    Available  Virtual Target Device - Logical Volume
ent3           Available  Shared Ethernet Adapter
```

Tip: It is useful to give the virtual device a name using the **-dev** flag with the **mkvdev** command for easier identification.

Slot numbering: Based on the **lsdev -vpd** command, the mappings exactly correspond to the slot numbering we intended (see Figure 12-1 on page 312). For example, the vhost0 device is slot number 20 (U9117.MMA.101F170-V1-C20) on the Virtual I/O Server, which is then being shared to the NIM_server partition. The NIM_server partition has its virtual SCSI device slot set to 21. This slot numbering is for easy association between virtual SCSI devices on the server and client side.

4. Use the **lsmap** command to ensure that all logical connections between newly created devices are correct, as shown in Example 12-9.

Example 12-9 Checking mappings

```
$ lsmap -all
SVSA          Physloc          Client
Partition ID
-----
vhost0        U9117.MMA.101F170-V1-C20    0x00000000

VTD           vnimsrv_rvg
Status        Available
LUN           0x8100000000000000
Backing device nimsrv_rvg
Physloc

SVSA          Physloc          Client
Partition ID
-----
vhost1        U9117.MMA.101F170-V1-C30    0x00000000

VTD           vdbsrv_rvg
Status        Available
LUN           0x8100000000000000
Backing device dbsrv_rvg
Physloc

SVSA          Physloc          Client
Partition ID
-----
```

vhost2	U9117.MMA.101F170-V1-C40	0x00000000										
<table> <tr> <td>VTD</td> <td>vIBMi_LS</td> </tr> <tr> <td>Status</td> <td>Available</td> </tr> <tr> <td>LUN</td> <td>0x8100000000000000</td> </tr> <tr> <td>Backing device</td> <td>IBMi_LS</td> </tr> <tr> <td>Physloc</td> <td></td> </tr> </table>			VTD	vIBMi_LS	Status	Available	LUN	0x8100000000000000	Backing device	IBMi_LS	Physloc	
VTD	vIBMi_LS											
Status	Available											
LUN	0x8100000000000000											
Backing device	IBMi_LS											
Physloc												
SVSA	Physloc	Client										
Partition ID												

vhost3	U9117.MMA.101F170-V1-C50	0x00000000										
<table> <tr> <td>VTD</td> <td>vlinux</td> </tr> <tr> <td>Status</td> <td>Available</td> </tr> <tr> <td>LUN</td> <td>0x8100000000000000</td> </tr> <tr> <td>Backing device</td> <td>linux</td> </tr> <tr> <td>Physloc</td> <td></td> </tr> </table>			VTD	vlinux	Status	Available	LUN	0x8100000000000000	Backing device	linux	Physloc	
VTD	vlinux											
Status	Available											
LUN	0x8100000000000000											
Backing device	linux											
Physloc												
SVSA	Physloc	Client										
Partition ID												

vhost4	U9117.MMA.101F170-V1-C90	0x00000000										
<table> <tr> <td>VTD</td> <td>vcd</td> </tr> <tr> <td>Status</td> <td>Available</td> </tr> <tr> <td>LUN</td> <td>0x8100000000000000</td> </tr> <tr> <td>Backing device</td> <td>cd0</td> </tr> <tr> <td>Physloc</td> <td>U789D.001.DQDYKYW-P4-D1</td> </tr> </table>			VTD	vcd	Status	Available	LUN	0x8100000000000000	Backing device	cd0	Physloc	U789D.001.DQDYKYW-P4-D1
VTD	vcd											
Status	Available											
LUN	0x8100000000000000											
Backing device	cd0											
Physloc	U789D.001.DQDYKYW-P4-D1											

Tips:

- The same concept applies when creating virtual disks that are going to be used as data volumes instead of boot volumes.
- You can map several disks through the same client-server adapter pair.

The mapped virtual disks will now appear to client partitions as generic SCSI disks, and the following chapter shows you how to create client partitions and install them.



Server virtualization implementation

This chapter shows you how to create and install the four client partitions for our basic Virtual I/O scenario shown in Figure 12-1 on page 312.

13.1 Creating a client partition

The client partition definitions are similar to the creation of our Virtual I/O Server partition, but instead of selecting **VIO Server**, choose **AIX or Linux**, or **IBM i**.

13.1.1 Procedure

Follow these steps to create the client partitions:

1. Restart the Create Logical Partition Wizard by selecting the server to create the logical partition on and choosing **Configuration** → **Create Logical Partition** with selecting either **AIX or Linux** or **IBM i** as shown in Figure 13-1.

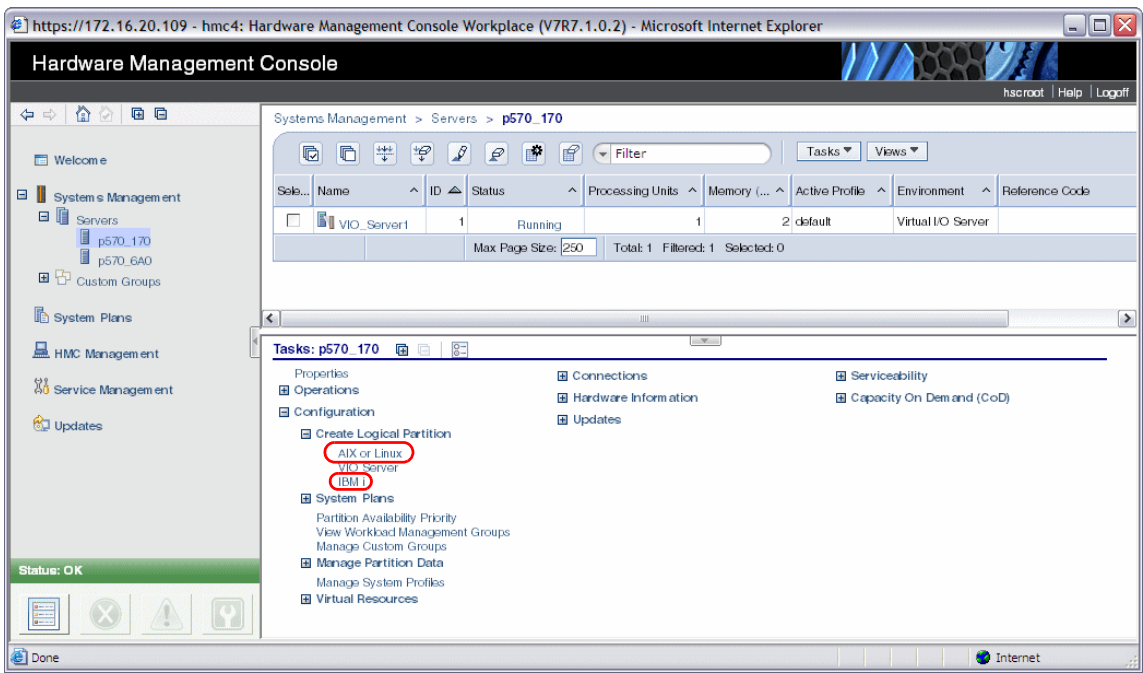


Figure 13-1 Creating client logical partition

2. Enter the name of the partition as shown in Figure 13-2. Note that Partition ID is 2. This ID was specified as the connecting client partition when the virtual SCSI server adapters were created on the Virtual I/O Server partition. After you have defined the Partition name, the HMC will update the definitions of the server SCSI adapters to add that Partition name.

https://172.16.20.109 - Create Lpar Wizard : p570_170 - Microsoft Internet Explorer

Create Lpar Wizard : p570_170

→ **Create Partition**

- Partition Profile
- Processors
- Processing Settings
- Memory Settings
- I/O
- Virtual Adapters
- Logical Host Ethernet Adapters (LHEA)
- Optional Settings
- Profile Summary

Create Partition

This wizard helps you create a new logical partition and a default profile for it. You can use the partition properties or profile properties to make changes after you complete this wizard.

To create a partition, complete the following information:

System name : p570_170

Partition ID : 2

Partition name : NIM_server

< Back Next > Finish Cancel Help

Done Internet

Figure 13-2 Create Partition dialog

3. Repeat steps 3 to 6 of 12.1, “Creating a Virtual I/O Server” on page 312 by choosing appropriate memory and processor values for your Virtual I/O Server client partition.
4. Click **Next** on the I/O dialog without selecting any physical I/O resources because we are not using physical adapters in our client partitions.
5. Create virtual Ethernet and SCSI client adapters. The start menu for creating virtual adapters is shown in Figure 13-3. The default serial adapters are required for console login from the HMC and must not be modified or removed.

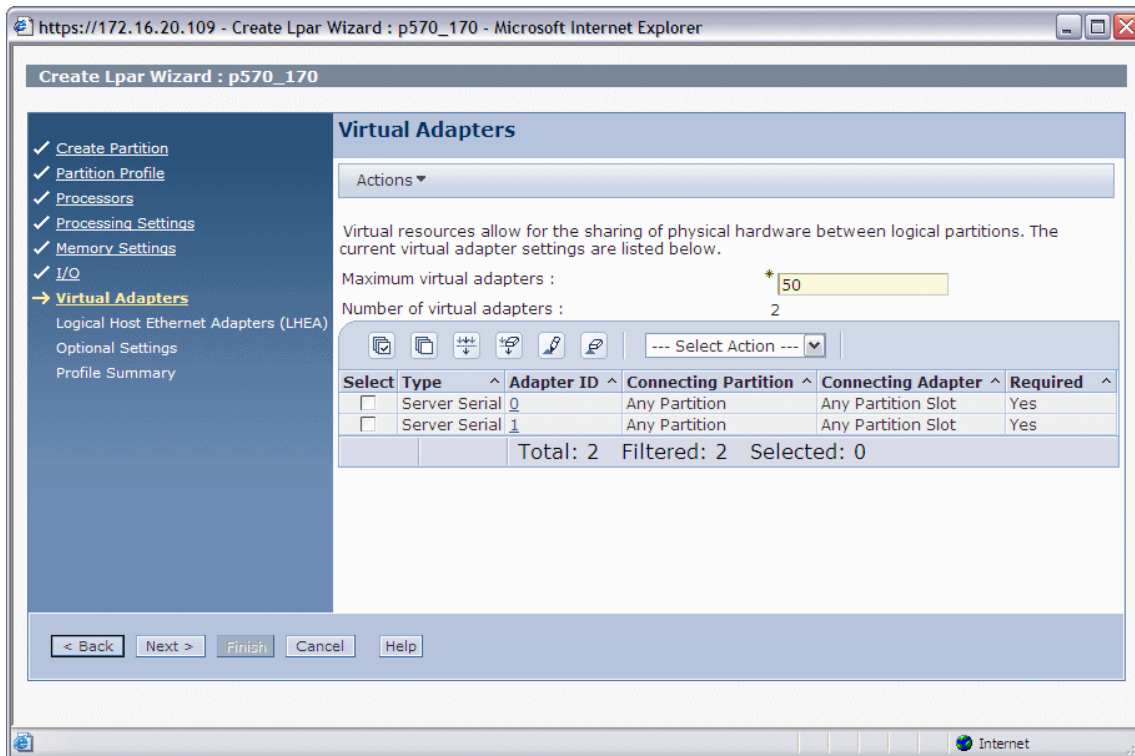


Figure 13-3 The start menu for creating virtual adapters window

Adapters: We increased the maximum number of virtual adapters to 50. Use any number that fits your configuration as long as it is less than 1024.

6. Select the drop-down menu path **Actions** → **Create** → **Ethernet Adapter** to open the Create Virtual Ethernet Adapter window. Create one virtual Ethernet adapter, as shown in Figure 13-4. Click **OK** when finished.

Important: Do not check the Use this adapter for Ethernet bridging box for client adapters.

The screenshot shows a dialog box titled "Virtual ethernet adapter". It has two tabs: "General" (selected) and "Advanced". Under "General", there are three input fields: "Adapter ID :" with the value "2", "VSwitch :" with a dropdown menu showing "ETHERNET0(Default)", and "Port Virtual Ethernet (VLAN ID):" with the value "1". To the right of the VLAN ID field is a button labeled "View Virtual Network...". Below these fields are three unchecked checkboxes: "This adapter is required for virtual server activation.", "IEEE 802.1q compatible adapter", and "Use this adapter for Ethernet bridging". The "IEEE Settings" section has a note: "Select this option to allow additional virtual LAN IDs for the adapter." The "Shared Ethernet Settings" section has a note: "Select Ethernet bridging to link (bridge) the virtual Ethernet to a physical network". At the bottom are three buttons: "OK", "Cancel", and "Help".

Figure 13-4 Creating a Client Ethernet adapter

7. Select the drop-down menu path **Actions** → **Create** → **SCSI Adapter** to open the Create Virtual SCSI Adapter window and create the virtual SCSI client adapter.

We want to create one SCSI adapter for disk and one SCSI adapter for the virtual optical device as shown in Figure 13-5 on page 358 and Figure 13-6 on page 358. Use Figure 12-1 on page 312 to select the correct client and server slot number. Make sure you select the correct Server partition in the menu. You can click **System VIOS Info** for more information about the slot numbers and their client-server relation.

Important: For IBM i, make sure to select **This adapter is required for partition activation** for the IBM i load source adapter, otherwise the IBM i client partition activation will fail.

Create the necessary adapters as shown in Figure 13-5 and Figure 13-6.

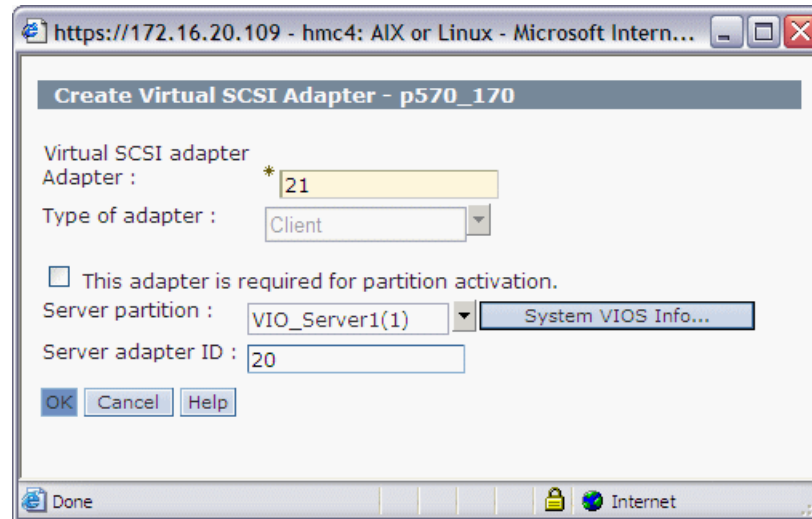


Figure 13-5 Creating the client SCSI disk adapter

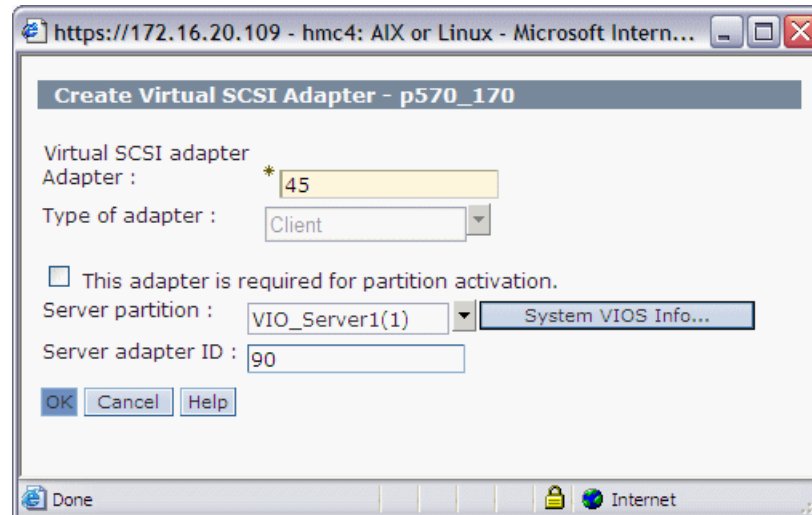


Figure 13-6 Creating the client SCSI DVD adapter

8. The list of created virtual adapters is shown in Figure 13-7. Click **Next** to continue.

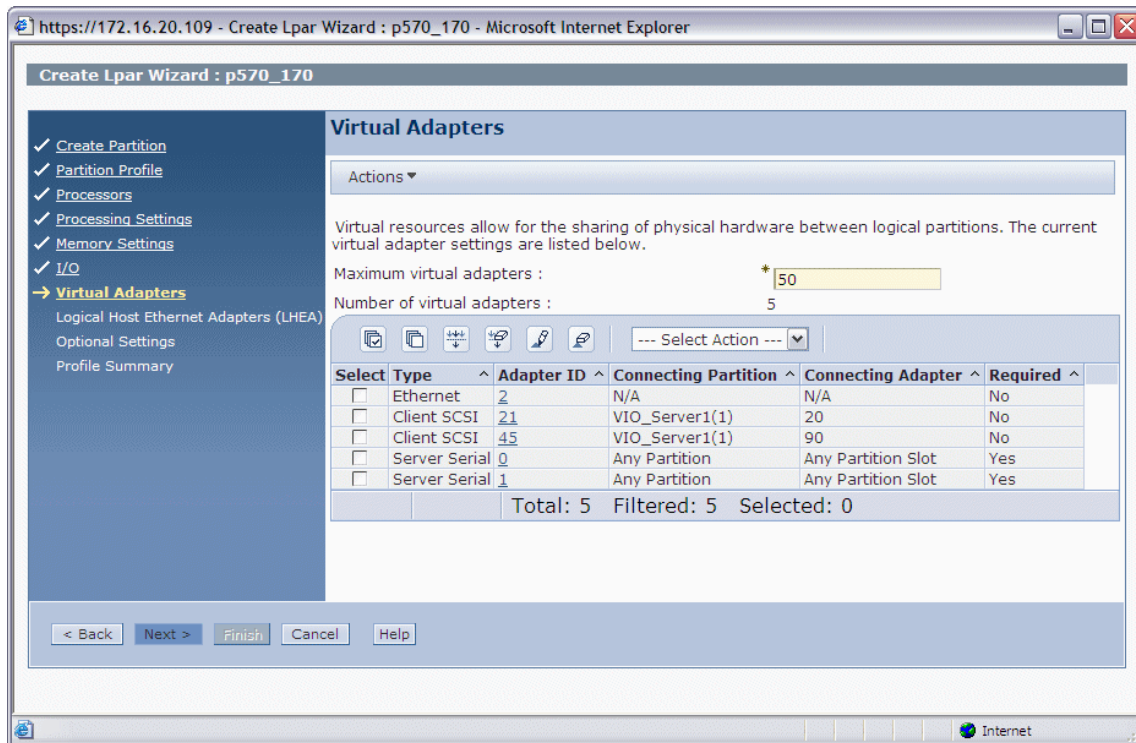


Figure 13-7 List of created virtual adapters

9. The Host Ethernet Adapter, HEA, is an obsolete feature on the POWER6 system. For information about HEA, see *Integrated Virtual Ethernet Adapter Technical Overview and Introduction*, REDP-4340 at:

<http://www.redbooks.ibm.com/abstracts/redp4340.html?Open>

We will not use any of these ports for our basic setup. The setup window is shown in Figure 13-8. Click **Next** to continue.

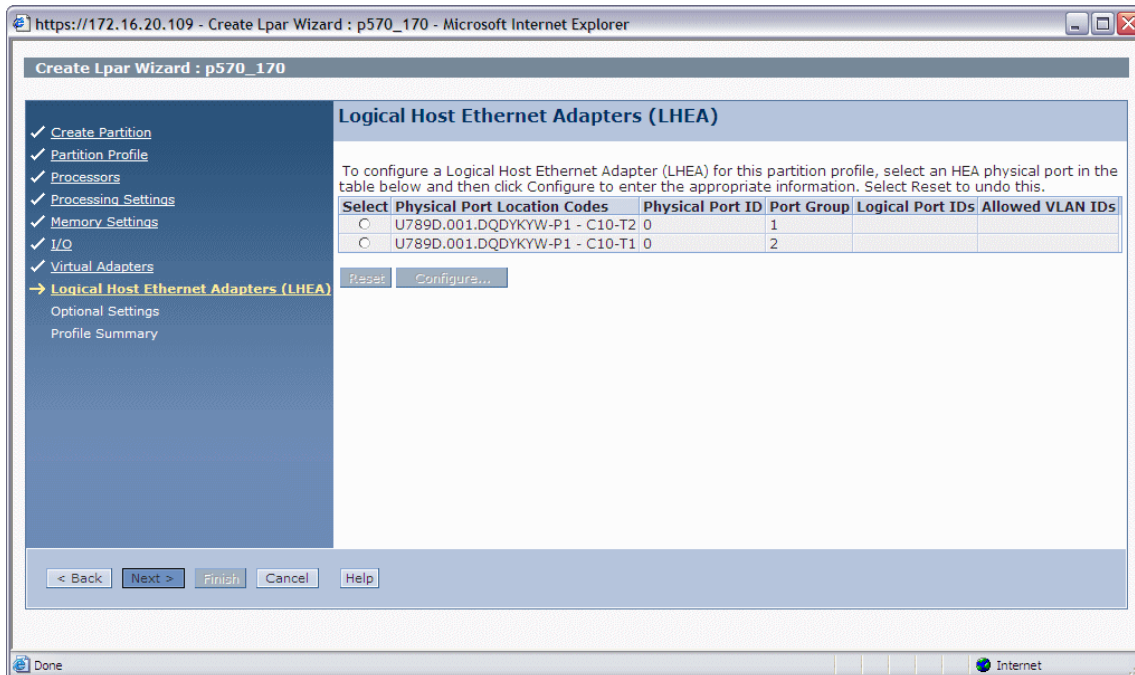


Figure 13-8 The Logical Host Ethernet Adapters menu

10. For an IBM i client partition only, optionally specify any OptiConnect settings in the OptiConnect Settings dialog and click **Next** to continue.
11. For an IBM i client partition only, specify the **Load source** and **Alternate restart device** (D-IPL device, in this example the virtual DVD device) adapter settings and optionally change the Console settings as shown in Figure 13-9.

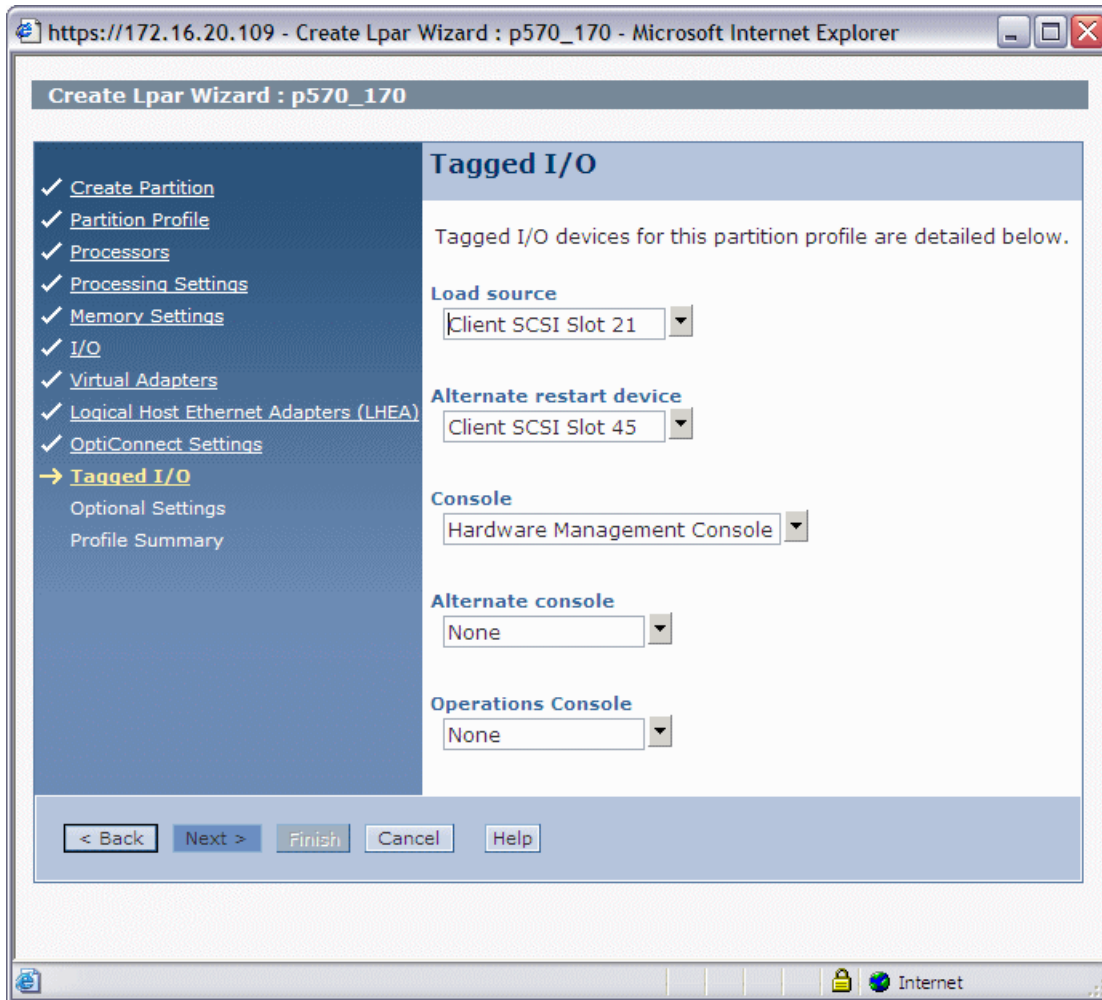


Figure 13-9 IBM i tagged I/O settings dialog

12. In the **Optional Settings** dialog of the Create LPAR Wizard (Figure 13-10), keep the default selection of **Normal** for boot modes and click **Next** to continue.

“Enable connection monitoring” will alert any drop in connection to the HMC. “Automatic start with managed system” means that the partition will start automatically when the system is powered on with the Partition auto start option (selected at power-on). “Enable redundant error path reporting” allows for call-home error messages to be sent through the private network in case of open network failure.

Important: Enable redundant error path reporting must *not* be set for partitions that will be moved using Partition Mobility.

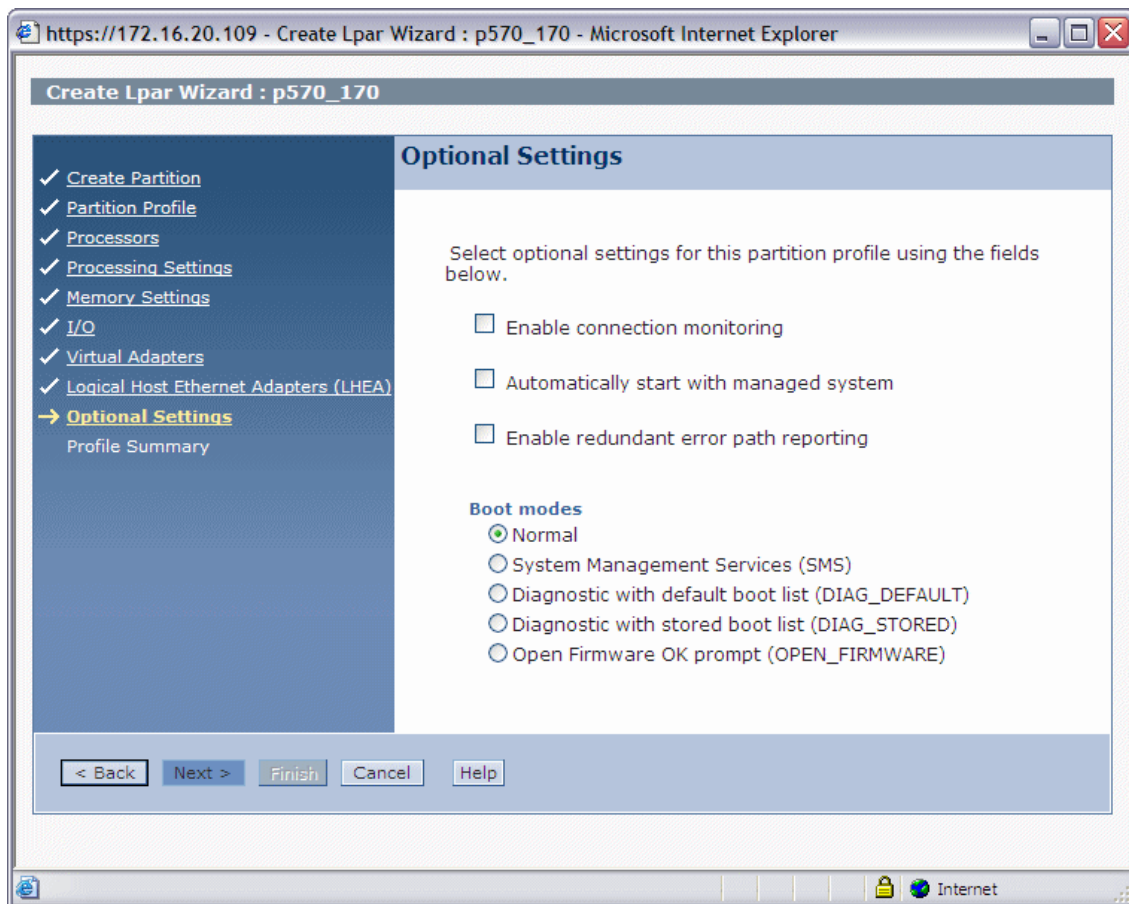


Figure 13-10 The Optional Settings menu

13. The Profile Summary menu (Figure 13-11) shows details about the partition. You can check details about the I/O devices by clicking **Details** or clicking **Back** to go back and modify any of the previous settings. Click **Finish** to complete the creation of the client partition.

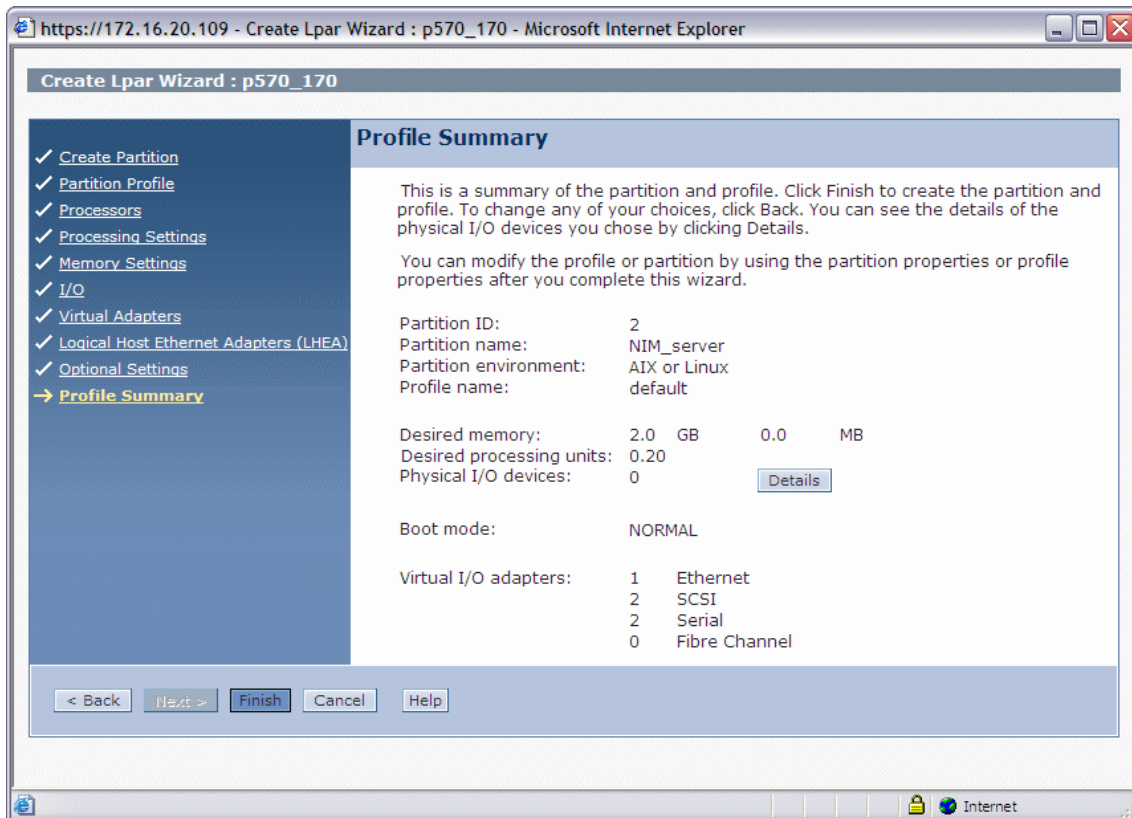


Figure 13-11 The Profile Summary menu

14. The complete list of partitions for the basic setup is shown in Figure 13-12. The partition *NIM_server* is selected for installation.

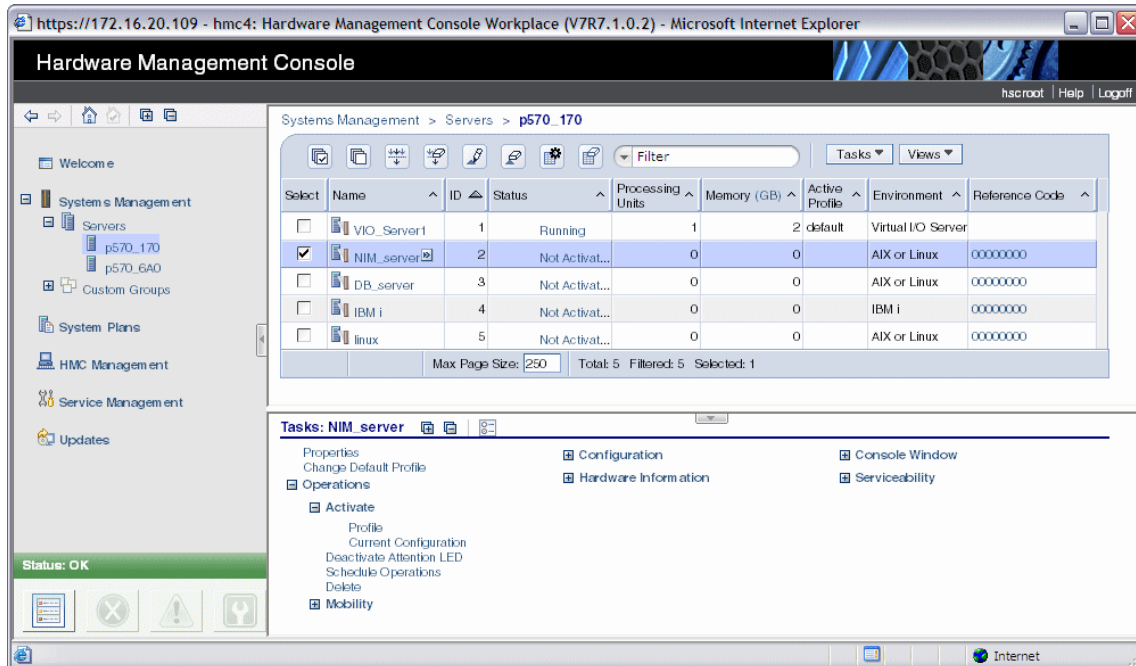


Figure 13-12 The list of partitions for the basic setup

15. It is best practice to back up the profile definition in case you want to restore it later. Activating the backup is shown in Figure 13-13. A menu is opened where you specify the name of the backup. Click **OK** to complete.

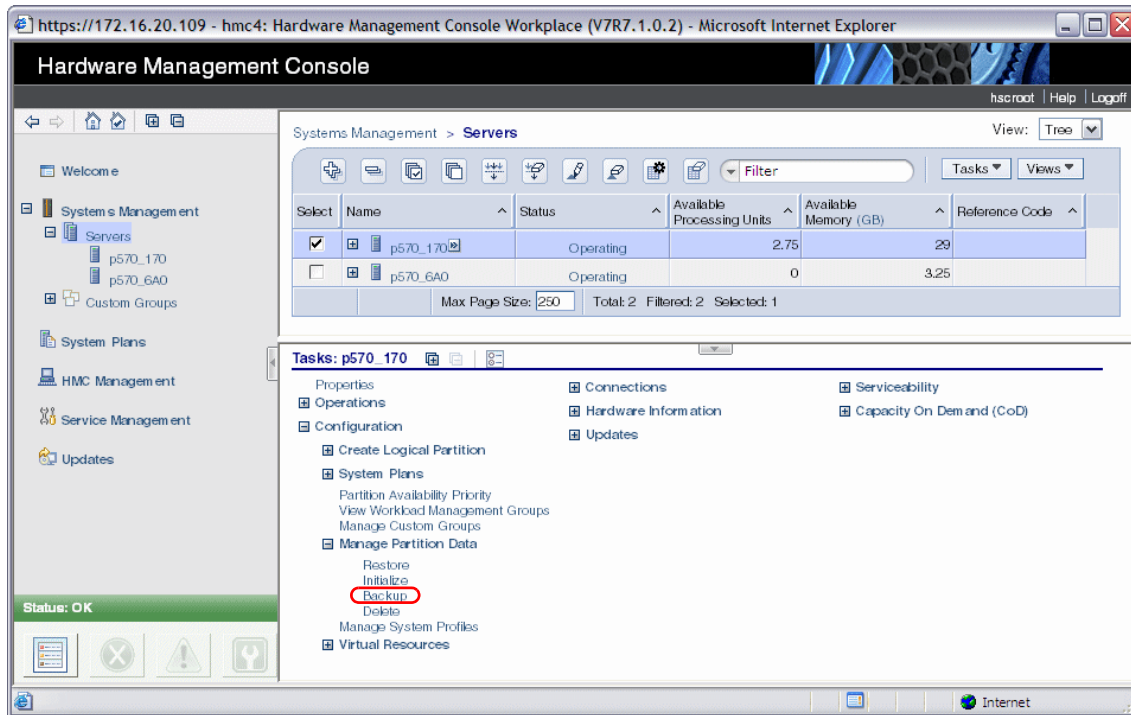


Figure 13-13 Backing up the profile definitions

13.1.2 Dedicated donating processors

When using dedicated processors, consider the options to donate unused processor cycles to Virtual Shared Processor Pools on POWER6 or later systems.

These options are set when editing the profile after it is created.

To set this option, do the following steps:

1. Select the partition with dedicated processors.
2. Select **Configuration** → **Manage Profiles** to open the Managed Profiles window.
3. Select a profile and click **Actions** → **Edit** as shown in Figure 13-14.

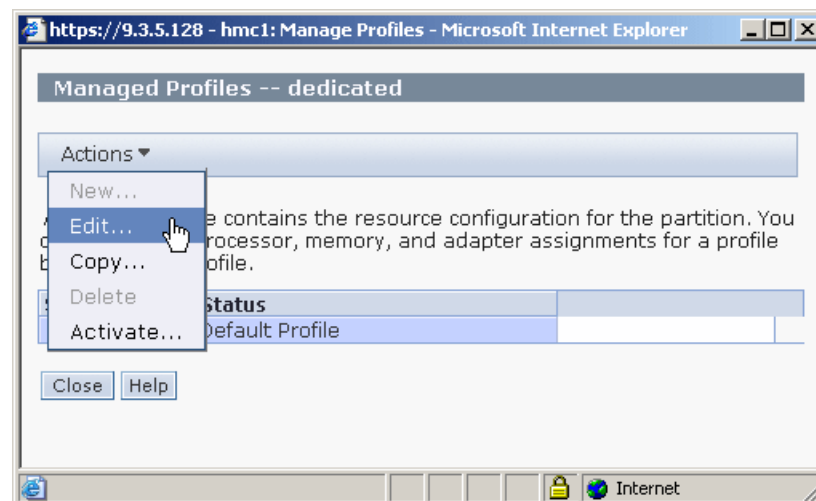


Figure 13-14 The edit Managed Profile window

4. The Logical Partition Profile Properties is opened. Open the Processors tab as shown in Figure 13-15 where the Processor Sharing options can be set.
 - The option **Allow when partition is inactive** is set by default and indicates whether the dedicated processors are made available to shared processor partitions when the logical partition that is associated with this partition profile is shut down.
 - The option **Allow when partition is active**, which is available on POWER6 systems or later, indicates whether the dedicated processors are made available to shared processor partitions when the logical partition that is associated with this partition profile is active.

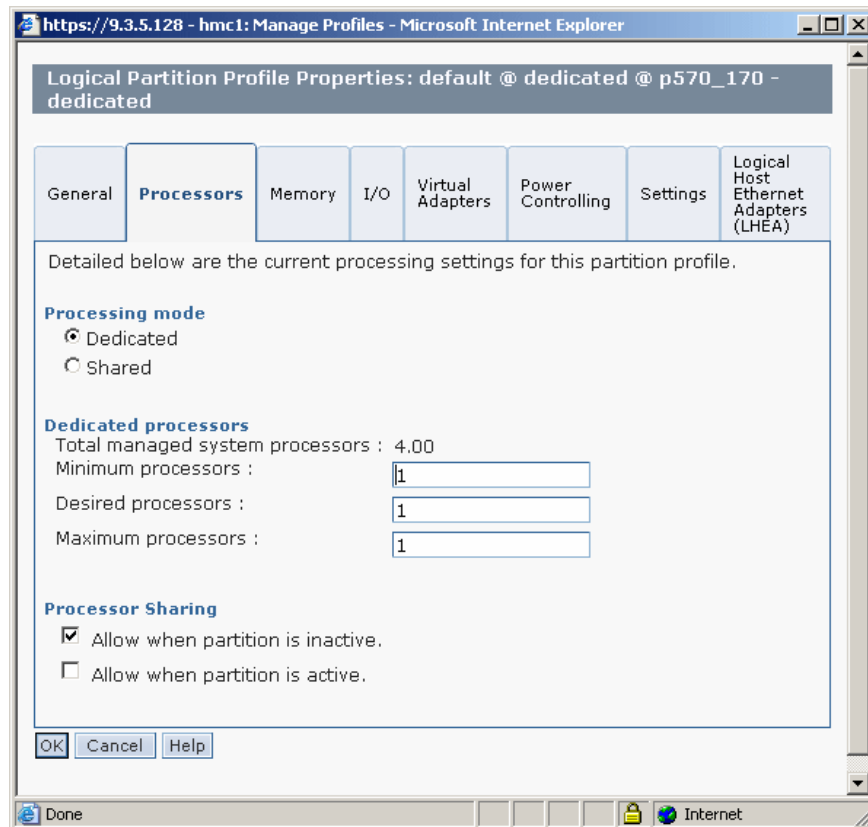


Figure 13-15 Setting the Processor Sharing options

Tip: You can get to the dialog shown in Figure 13-15 while a partition is running by clicking **LPAR properties** to change this dynamically.

13.2 AIX client partition installation

This section describes the method to install AIX onto a previously defined client partition. You can choose your preferred method, but for our basic scenario, we opted to install the NIM_server from CD and then install the DB_server partition using the Network Installation Manager (NIM) from the NIM_server. We are also going to use the virtual Ethernet adapters for network booting and the virtual SCSI disks that were previously allocated to client partitions for rootvg.

Tip: A virtual optical device can be used for a CD or DVD installation as long as it is not already assigned to another client partition.

Assuming that a NIM master is configured, the following basic steps are required to perform an AIX installation using NIM:

1. Create the NIM machine client dbserver and definitions on your NIM master. Example 13-1 shows how to check for allocated resources.

Example 13-1 Check if resources had been allocated

```
# lsnim -l dbserver
dbserver:
  class          = machines
  type           = standalone
  connect        = nimsh
  platform       = chrp
  netboot_kernel = 64
  if1            = network1 dbserver 0
  cable_type1    = N/A
  Cstate         = BOS installation has been enabled
  prev_state     = ready for a NIM operation
  Mstate         = not running
  boot           = boot
  lpp_source     = aix61_lppsource
  nim_script     = nim_script
  spot           = aix61_spot
  control        = master
```



```
# tail /etc/bootptab
#      T170 -- (xstation only) -- server port number
#      T175 -- (xstation only) -- primary / secondary boot host
indicator
#      T176 -- (xstation only) -- enable tablet
#      T177 -- (xstation only) -- xstation 130 hard file usage
#      T178 -- (xstation only) -- enable XDMCP
#      T179 -- (xstation only) -- XDMCP host
#      T180 -- (xstation only) -- enable virtual screen
dbserver:bf=/tftpboot/dbserver:ip=9.3.5.113:ht=ethernet:sa=9.3.5.197
:sm=255.255.254.0:
```

2. Initiate the install process by activating the DB_server client partition in SMS mode, as shown in Figure 13-16.

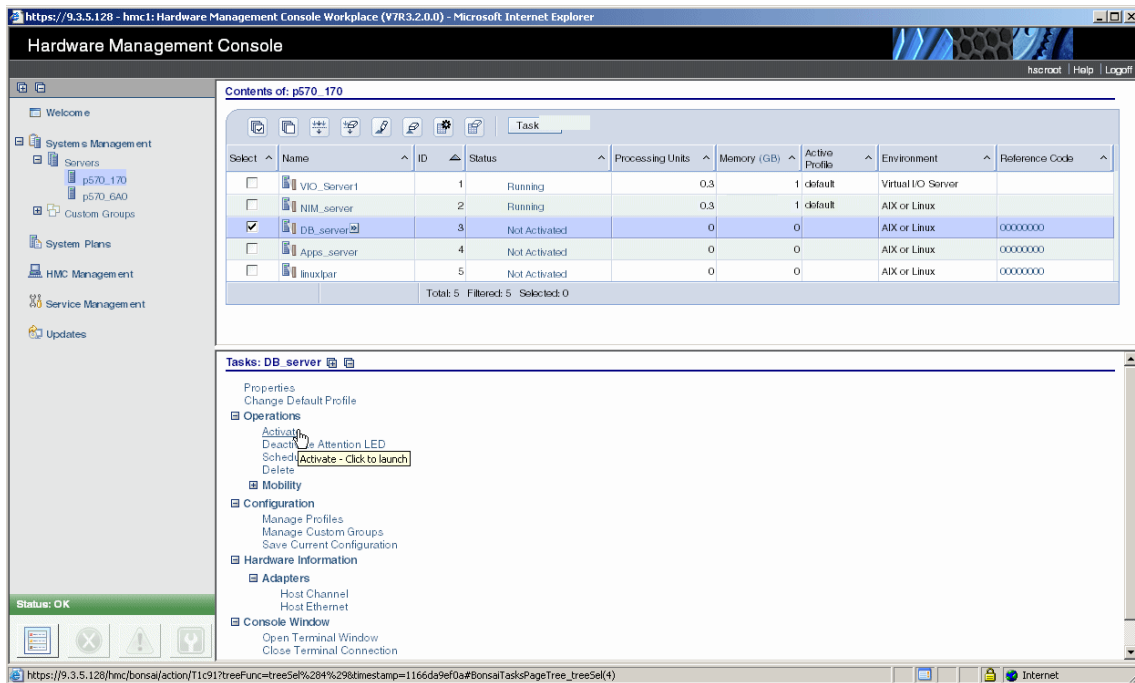


Figure 13-16 Activating the DB_server partition

3. Set up the network boot information by choosing option **2, Setup Remote IPL** (see Figure 13-17).

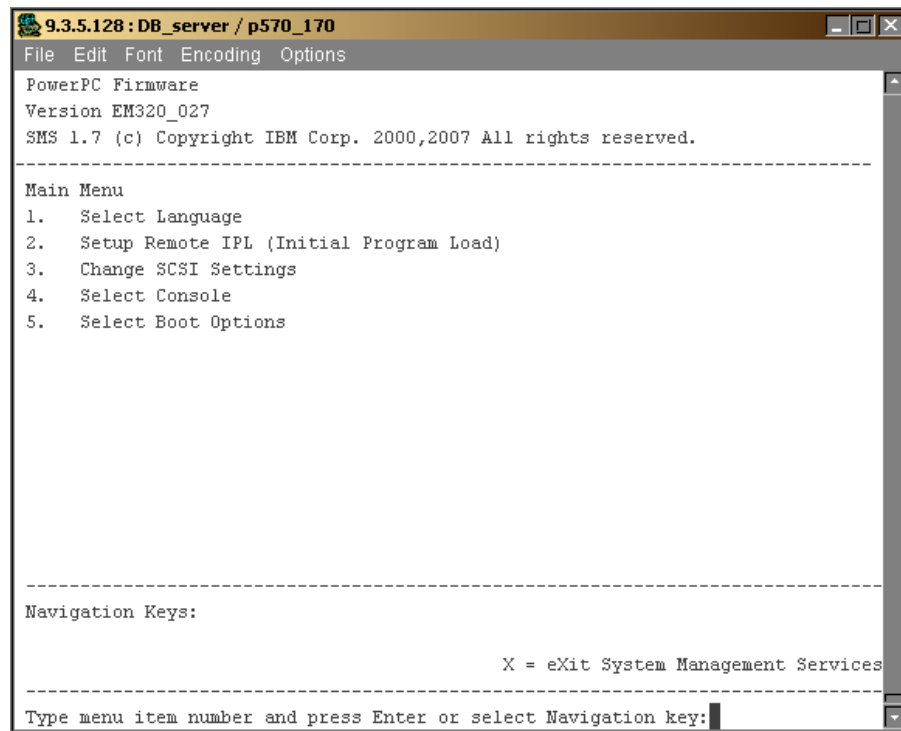


Figure 13-17 The SMS menu

4. Choose option **1**, as shown in Figure 13-18.

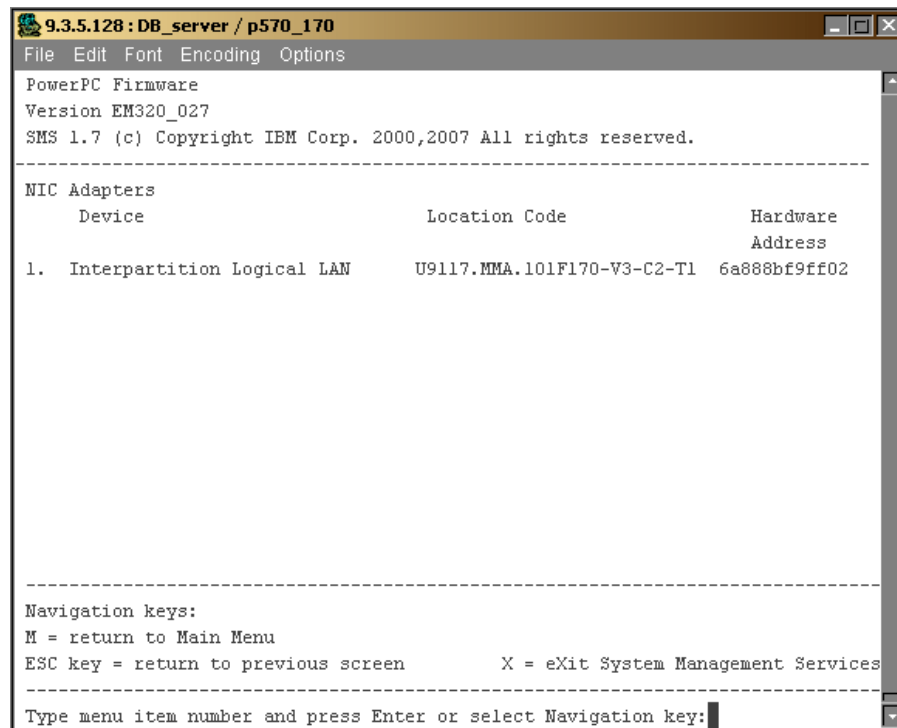


Figure 13-18 Selecting the network adapter for remote IPL

Tip: Interpartition Logical LAN number 1 is the virtual Ethernet adapter that was defined on the NIM master for the DB_server client:

```
dbserver:
  class      = machines
  type       = standalone
  connect    = nimsh
  platform   = chrp
  netboot_kernel = 64
  if1        = network1 dbserver 0
```

5. Choose option **1** for IP Parameters, then go through each of the options and supply the IP address, as shown in Figure 13-19.

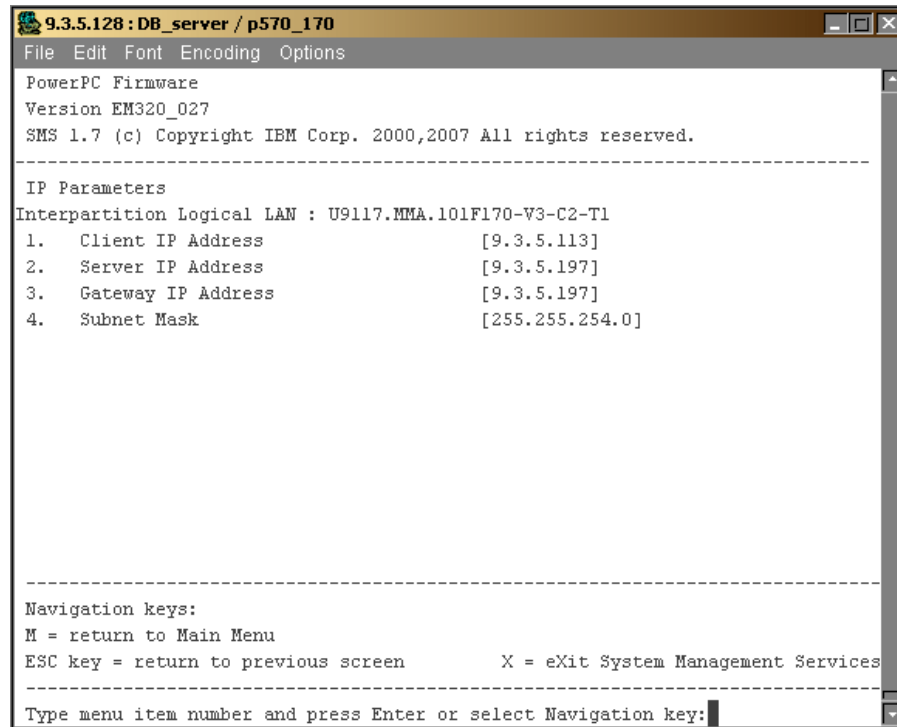


Figure 13-19 IP settings

6. Press Esc to go one level up for the **Ping test** as shown in Figure 13-20.

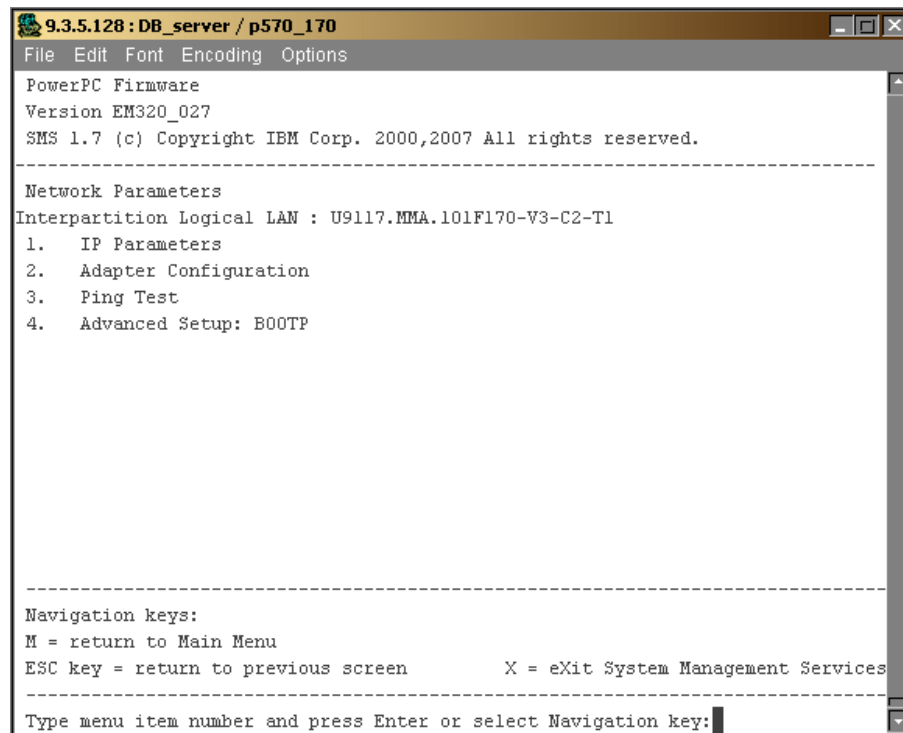


Figure 13-20 Ping test

7. Select **3** to execute a ping test and, provided it is successful, you are ready to do the NIM installation.
8. Press **M** to get back to the main menu.
9. Select 5 to verify that boot is set to the correct network adapter.
10. Select 2 for boot order.
11. Select the first boot device, option 1.

12. It is useful to list all devices with option **8**. In Figure 13-21 you can see that no device is listed in Current Position.

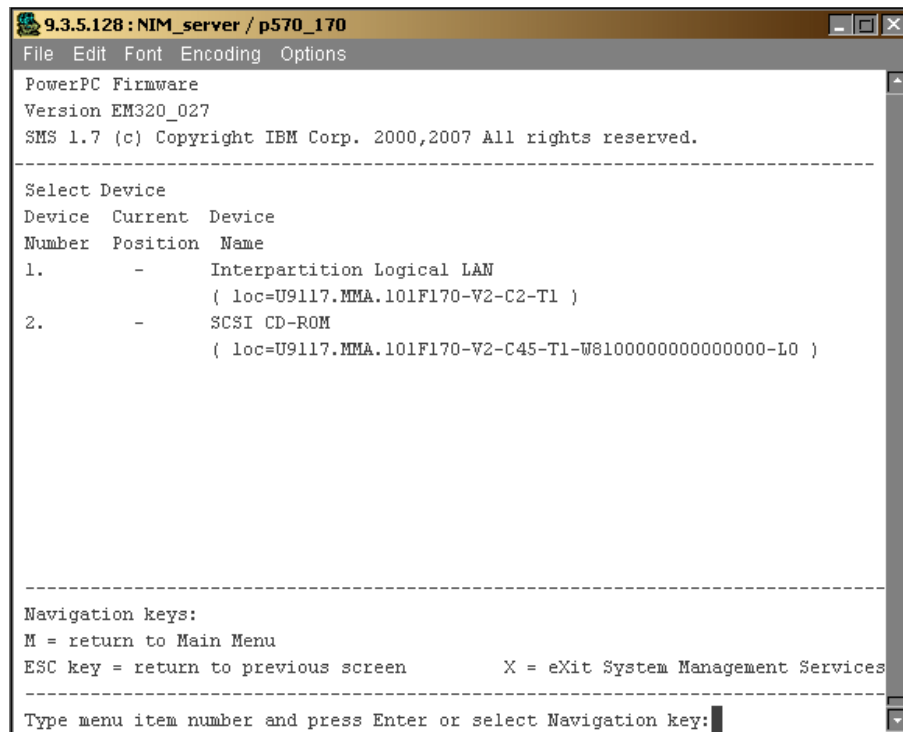


Figure 13-21 Setting the install device

13. Select option **1**. We want the Interpartition Logical LAN to be our boot device.

14. Select option **2** to Set Boot Sequence.

15. Click **X** to exit from SMS and confirm with **1** to start the network boot.

Tip: It is best practice to use separate volume groups for applications and user data in order to keep the rootvg volume group reserved for the AIX operating system. This makes rootvg more compact and easier to manipulate if required.

13.3 Installation of IBM i client partition

This section discusses the installation of an IBM i client partition.

13.3.1 Overview

IBM i client partition can be setup and manage by Hardware Management Console (HMC) or IBM Integrated Virtualization Manager (IVM).

Notice the difference between client logical partition and logical partaken.

Client logical partition A partition that is using some or all of its I/O resources from another partition; for example, IBM i using the resources of the Virtual I/O Server on a POWER system.

Logical partition A partition that is using its own physical resources

Table 13-1 summarizes the environments in which IBM i can be a client logical partition.

Table 13-1 IBM i client logical partition environments

System Hardware	Management Tool	Server logical partition	Client logical partition
POWER 6 or later processor-based blade server	Integrated Virtualization Manager	Virtual I/O Server	IBM i
POWER 6 or later processor-based server	Integrated Virtualization Manager	Virtual I/O Server	IBM i
POWER 6 or later processor-based server	Hardware Management Console	Virtual I/O Server	IBM i
POWER 6 or later processor-based server	Hardware Management Console	IBM i	IBM i

13.3.2 Considerations for IBM i client partitions managed by IVM

The following limitations and restrictions apply to IBM i client logical partitions of the Virtual I/O Server that are running on systems that are managed by the Integrated Virtualization Manager:

- ▶ IBM i client logical partitions do not own any physical I/O resources. All I/O resources on the IBM i client partition are virtual Ethernet and virtual storage (disk and optical).
- ▶ IBM i client logical partition has no view of the physical hardware. This affects servicing of the physical hardware and also the operation and amount of data returned by some commands, APIs, and MATATR machine instruction.
- ▶ A minimum of 1 GB of memory is recommended for Virtual I/O Server with an IBM i client partition on a blade or POWER processor-based system.

Note: Although you can run the IBM i client partition with only virtual I/O hardware on HMC-managed systems, you still have the option to assign physical hardware to the IBM i client partition for any function that are not supported by virtual I/O hardware.

13.3.3 Installation considerations

The best way to install IBM i is from a physical DVD drive owned by the Virtual I/O Server. Use Integrated Virtualization Manager to virtualize and assign the optical drive, which is physically owned by the Virtual I/O Server, to IBM i. Then you can access the DVD drive virtually from the IBM i partition.

If the Virtual I/O Server does not have a physical DVD drive, you can upload the IBM i install media to a virtual optical file in the Virtual I/O Server. Because the installation media for IBM i is greater than 2 GB, you must use the Virtual I/O Server command line to upload the IBM i installation media to the Virtual I/O Server.

13.3.4 Installation process

The IBM i installation process for using virtual devices is the same as for native attached storage devices.

Before activating the IBM i virtual I/O client partition using a D-IPL manual mode for installing IBM i, ensure that the following requirements are met:

- ▶ The load source is tagged to the virtual SCSI or virtual Fibre Channel adapter mapped on the Virtual I/O Server for the IBM i client partition.

- The alternate restart device is tagged to the virtual SCSI optical or virtual tape device or else to a native attached restart device such as a physical tape.

As with any IBM i installation on a new partition, after selecting the Install Licensed Internal Code option, the Select Load Source Device screen is shown as in Figure 13-22.

Notice that virtual SCSI disk devices are shown with a generic type 6B22 and model 050. The Sys Card information shows the virtual SCSI client adapter ID as defined in the partition profile. The Ctl information XOR 0x80 corresponds to the virtual target device LUN information as shown in the Virtual I/O Server's **lsmap -all** command output, for example, Ctl 1 corresponds to LUN 0x81.

Select Load Source Device

Type 1 to select, press Enter.

Opt	Serial Number	Type	Model	Sys Bus	Sys Card	I/O Adapter	I/O Bus	Ctl	Dev
	YAP8GVNPCU7Z	6B22	050	255	21	0	0	4	0
	Y8VG3JUGRKLD	6B22	050	255	21	0	0	2	0
1	Y9UCTLXBVQ9G	6B22	050	255	21	0	0	1	0
	YW9FPXR5X759	6B22	050	255	21	0	0	3	0

F3=Exit
F5=Refresh
F12=Cancel

Figure 13-22 IBM i Select load source device panel

For further information about installing IBM i on a new logical partition, see the IBM i Information Center at this website:

<http://publib.boulder.ibm.com/infocenter/iserics/v7r1m0/index.jsp?topic=/rzahc/rzahcinstall.htm>

13.4 Linux client partition installation

There are several ways to install Linux distributions on Power systems:

- ▶ Attended or unattended
- ▶ Network-based or media-based

The following sections describe the available tooling for you to decide the way that best fits your needs. They also show how to start a Linux installation on a client partition, either from the network or from a Virtual Media Library on the Virtual I/O Server. The installation from a Virtual Media Library is faster than from a physical DVD media and it is simple to set up.

13.4.1 Unattended installations

A Linux installation can be fully automated and run unattended using the IBM Installation Toolkit or Linux distribution-specific tools such as autoyast or kickstart.

IBM Installation Toolkit for Linux is available as a bootable ISO image which starts a Live DVD environment accessible through a friendly browser or text user interface. IBM Installation Toolkit provides the following facilities:

- ▶ Automatic installation of IBM Service and Productivity Tools, including dynamic LPAR functionality.
- ▶ Automatic setup of IBM POWER Linux Tools Repository, which enables the use of standard Linux package management tools to provide access to IBM Service and Productivity Tools, IBM Software Development Toolkit, and IBM Advance Toolchain for POWER Linux servers.
- ▶ Auxiliary wizards for these tasks:
 - System firmware updates
 - Deployment, configuration, and maintenance of a Linux installation server, which can host Linux distribution and IBM Installation Toolkit repositories for multiple and future installations
 - Migration of LAMP software stack from a x86 server to a POWER system.

LAMP: The solution stack composed by Linux, Apache HTTP Server, MySQL database and PHP, Perl, or Python.

For more information on IBM Installation Toolkit such as latest updates, download links, supported Linux distributions, and to get access to complete information, see this website:

<http://www14.software.ibm.com/webapp/set2/sas/f/lopdiags/installtools>

When using IBM Installation Toolkit, kickstart or autoyast files are not necessary. The kickstart or autoyast files are used to start an unattended Linux installation through distribution-specific tools. For more information on how to use Linux distribution-specific tools such as autoyast or kickstart, check their respective documentation.

13.4.2 Multipath devices and Linux installations

Starting with IBM Installation Toolkit version 5.3, Linux installation on multipath devices is implemented for RHEL 6, SLES 10, and SLES 11. IBM Installation Toolkit does not support installing RHEL 5 on multipath devices.

When installing RHEL 5 and older releases on multipath devices through Linux distribution media, the dm-multipath driver must be loaded with the kernel boot prompt parameter *mpath*.

Important: You might have configuration issues if performing an RHEL 5 installation on a single path device and configuring multipath in a later stage for the installed device.

Loading the dm-multipath driver is not necessary to perform RHEL 6 and SLES multipath installations using either Linux distribution media or IBM Installation Toolkit. For SLES 10 multipath installations using Linux distribution media, see this website:

http://www.novell.com/documentation/sles10/stor_admin/?page=/documentation/sles10/stor_admin/data/mpiotools.html

13.4.3 Starting a Linux installation from the network

Linux network installation on POWER requires the following components:

- ▶ BOOTP or DHCP server to answer BOOTP requests
- ▶ TFTP server to provide the boot file
- ▶ NFS, FTP, or HTTP server to provide the installation files repository
- ▶ Optional kickstart or autoyast file for unattended automated installation without IBM Installation Toolkit

These components are usually located on the same server. The installation server may run AIX (for example, a NIM server) or Linux.

A Linux server can be easily configured through IBM Installation Toolkit wizards. If the server was configured without DHCP, a Linux network installation can be started by configuring IPL settings on SMS menu to point to that Linux server. With DHCP, the installation is started by just booting the server from network. There is no need to configure IPL settings on SMS. If they are already configured, they must be zeroed.

The following sections show alternative ways to set up an AIX and a Linux server, and start a Linux network installation from them. They also show how to start a Linux installation from a Virtual Media Library device.

Using an AIX server for starting a Linux installation

If you plan to use an AIX server for installing Linux, it must have BOOTP, TFTP and NFS enabled. This is normally already the case if the AIX server is used as a NIM server. If not, you have to configure these services in `/etc/inetd.conf` and refresh the `inetd` daemon.

The following steps are required to enable a Linux network installation from an AIX server:

1. Add the following line to the `/etc/bootptab` file. Adapt the IP addresses and subnet mask according to your environment:

```
linuxlpar0:bf=/tftpboot/ppc64.img:ip=9.3.5.200:ht=ethernet:sa=9.3.5.197:sm=255.255.254.0:
```

If you are going to initiate a broadcast BOOTP request, add the MAC address to the line using the `ha=XXXXXXXXXXXX` statement. The `ha` statement must be placed after the `ht=ethernet` statement.

2. Copy the content of the installation DVDs to the installation server, then export the directory by adding it to the `/etc/exports` file as shown here:

```
/export/linux/rhel  
/export/linux/sles
```

Tip: If you are getting “permission denied” errors when trying to NFS mount, add the IP address of the Linux partition that you are trying to install to the `/etc/hosts` file on the installation server.

3. Copy the network boot kernel to /etc/tftpboot. For SLES 10 the file is called inst64. For RHEL 5 the file is called ppc64.img. Starting with RHEL 6 the netboot installation procedures have changed. Due to the increase in the size of the install image and the limited amount of real memory available to the installer, Yaboot must be used to perform network installations.

Yaboot: RHEL 6 installations use Yaboot search for the configuration file *tftpboot/etc/yaboot.conf* to determine where to get the installation image. For multiple installations, create the Yaboot configuration files in the *01-<mac_address>* format.

4. Set up the remote IPL settings in SMS or use the Open Firmware prompt to perform a network boot. The Open Firmware prompt is only needed to provide boot arguments for the RHEL 5 installation for example, to enable multipath or to make VNC available during the installation. The following example shows how to perform a network boot from the Open Firmware prompt:

```
0 > devalias net /vdevice/l-1an@300000002 ok
0 > boot net:9.3.5.197,,9.3.5.115, install mpath vnc
vncpassword=abc123
```

Boot: RHEL 6 does not require the mpath boot parameter to install on a multipath device. The dm-multipath driver is automatically loaded and the devices can be selected in the graphical interface.

Using a Linux server for starting a Linux installation

If you decide not to use IBM Installation Toolkit and have a Linux installation server with DHCP enabled, it can be used to start the installation of Linux partitions.

Add the following lines to /etc/dhcpd.conf:

```
ignore unknown-clients;
not authoritative;
allow bootp;
allow booting;
```

Attention: The DHCP server only answers to BOOTP broadcast requests. Therefore, the IPL settings defined in SMS must not contain any IP address settings. If you set the IP address in order to test connectivity to the installation server using the ping functionality in SMS, make sure that you reset the values back to zero before initiating the installation.

13.4.4 Starting a Linux installation from Virtual Media Library

An alternative to the network installation is to create a Virtual Media Library device on the Virtual I/O Server and load Linux distribution or IBM Installation Toolkit image files on the virtual device.

To set up a Virtual Media Library device, follow these steps:

1. Add one Virtual SCSI Adapter to the Virtual I/O Server partition profile. Also add a Virtual SCSI adapter to the Linux client partition and map the client adapter to the server adapter number.
2. On the Virtual I/O Server, create the Virtual Media Library device with the following command:
3. Download Linux distribution image and run the **mkvopt** command to create a virtual optical media:

```
$ mkvdev -fbo -vadapter vhostX
```

```
$ mkvopt -ro -name linux -file /home/padmin/linux.iso
```

IBM Installation Toolkit: Some POWER systems come with a preloaded IBM Installation Toolkit image on its Virtual Media Library. Alternatively, you can download the latest IBM Installation Toolkit version and manually create an IBM Installation Toolkit virtual optical media through the same **mkvopt** command.

Read-only: The **-ro** flag creates a read-only virtual optical media and allows it to be assigned to more than one logical partition at a time.

4. Load the Linux distribution (or IBM Installation Toolkit) image to the virtual device:
5. Activate the Linux partition and set the option on SMS menus to boot from the optical device.

```
$ loadopt -vtd vtoptX -disk linux
```

After you complete the installation wizard, IBM Installation Toolkit will ask for the Linux distribution media. In that case, for a completely unattended Linux installation, you can use two Virtual Media Library devices to load both the IBM Installation Toolkit and Linux distribution media at the same time.

With only one Virtual Media Library, it will be necessary to unload IBM Installation Toolkit image and load Linux distribution media. To unload the image from the Virtual Media Library device, run the following command on the Virtual I/O Server:

```
$ unloadopt -vtd vtoptX
```

13.4.5 Installing IBM service and productivity tools

After the initial installation, if IBM Installation Toolkit was not used, additional service and productivity tools provided by IBM must be installed. These tools are required for the dynamic LPAR functionality. You can find the tools and the installation instructions at this website:

<https://www14.software.ibm.com/webapp/set2/sas/f/lopdiags/home.html>



Part 4

Set up

This part of the book shows how to set up the initial virtualization, including high availability configurations.

This part includes the following topics:

- ▶ Processor virtualization setup
- ▶ Memory virtualization setup
- ▶ I/O virtualization setup
- ▶ Server virtualization setup

For advanced setup and maintenance information, see *IBM PowerVM Virtualization Managing and Monitoring*, SG24-7590.



Processor virtualization setup

This chapter shows the basics of setting up Multiple Shared-Processor Pools and shared-dedicated capacity.

Before reading this chapter, you need to be familiar with the concepts and terms described in Chapter 2, “Processor virtualization overview” on page 19.

14.1 Configuring Multiple Shared-Processor Pools

All Power Systems that support the Multiple Shared-Processor Pools capability will have a minimum of one (the default) Shared-Processor Pool and up to a maximum of 64 Shared-Processor Pools.

The default Shared-Processor Pool (SPP₀) is automatically activated by the system and is always present. Its Maximum Pool Capacity is set to the capacity of the Physical Shared-Processor Pool. For SPP₀, the Reserved Pool Capacity is always 0.

All other Shared-Processor Pools exist, but by default, are inactive. By changing the Maximum Pool Capacity of a Shared-Processor Pool to a value greater than zero, it becomes active and can accept micro-partitions (either transferred from SPP₀ or newly created).

The system administrator can use the HMC to activate additional Multiple Shared-Processor Pools. As a minimum, the Maximum Pool Capacity will need to be specified. If you want to specify a Reserved Pool Capacity, there must be enough unallocated physical processor capacity to guarantee the entitlement.

Terminology: The terms used in Chapter 8, “Processor virtualization planning” on page 111, are not the same as you will see on the HMC. The Maximum Pool Capacity (MPC) is called *maximum processing units* in the HMC dialogs. Reserved Pool Capacity (RPC) is referred to as *reserved processing units*. In the text of the following sections, the terms as they are defined in Chapter 8, “Processor virtualization planning” on page 111 are used. On the screen captures, you will see the HMC terminology.

14.1.1 Dynamic adjustment of Maximum Pool Capacity

The Maximum Pool Capacity of a Shared-Processor Pool, other than the default Shared-Processor Pool₀, can be adjusted dynamically from the HMC using either the graphical or CLI interface.

14.1.2 Dynamic adjustment of Reserve Pool Capacity

The Reserved Pool Capacity of a Shared-Processor Pool, other than the default Shared-Processor Pool₀, can be adjusted dynamically from the HMC using either the graphical or CLI interface.

14.1.3 Dynamic movement between Shared-Processor Pools

A micro-partition can be moved dynamically from one Shared-Processor Pool to another from the HMC using either the graphical or CLI interface. As the Entitled Pool Capacity is partly made up of the sum of the entitled capacities of the micro-partitions, removing a micro-partition from a Shared-Processor Pool will reduce the Entitled Pool Capacity for that Shared-Processor Pool. Similarly, the Entitled Pool Capacity of the Shared-Processor Pool that the micro-partition joins will increase.

14.1.4 Deleting a Shared-Processor Pool

Shared-Processor Pools cannot be deleted from the system. However, they are deactivated by setting the Maximum Pool Capacity and the Reserved Pool Capacity to zero. The Shared-Processor Pool will still exist but will not be active. Use the HMC interface to deactivate a Shared-Processor Pool. A Shared-Processor Pool cannot be deactivated unless all micro-partitions hosted by the Shared-Processor Pool have been removed.

14.1.5 Configuration scenario

This section describes how to create Multiple Shared-Processor Pools to limit the amount of processing capacity that can be consumed by the micro-partitions based on the micro-partitions entitlement, virtual processors, pool name, and pool ID described in Table 14-1. In this scenario, all micro-partitions are running in uncapped mode.

Table 14-1 Micro-partition configuration and Shared-Processor Pool assignments

Partition name	Entitled Capacity	Number of virtual processors	Pool name	Pool ID
VIO_Server1	0.3	2	DefaultPool	0
VIO_Server2	0.3	2	DefaultPool	0
DB_server	1	3	Production	1
Apps_server	1	3	Production	1
NIM_server	0.2	1	Development	2
linuxlpar	0.2	1	Development	2

The two Virtual I/O Servers are created in the default Shared-Processor Pool (Pool ID = 0). On the HMC this pool is referred to as DefaultPool. DefaultPool cannot be modified. Each of the Virtual I/O Servers has an entitlement of 0.3 processing units and 2 virtual processors configured. Each one is therefore guaranteed to get 0.3 processing units and can consume up to 2 processing units if required and if spare processing capacity is available in the DefaultPool.

The DB_server and the Apps_server micro-partitions are assigned to the Shared-Processor Pool called Production (Pool ID = 1). As shown in Table 14-2, the Maximum Pool Capacity of the Production pool is limited to 3. A reason for limiting a production workload might be to reduce licensing cost for applications running in these partitions.

Table 14-2 Shared-Processor Pool attributes

Pool name	Maximum Pool Capacity	Entitled Pool Capacity	Reserved Pool Capacity
Production	3	2.5	0.5
Development	1	0.4	0

Tip: The pool name can be up to 14 characters long and can contain blanks.

The Entitled Pool Capacity, which is the amount of processing units that the pool is guaranteed to get, is 2.5 for the Production pool. This is the sum of the entitlements of the partitions in the pool (in this case 1.0 for the DB_server and 1.0 for the Apps_server) plus the Reserved Pool Capacity of 0.5 processing units. The Reserved Pool Capacity is distributed within the bounds of the pool according to the weighting factors defined for each partition. A reason for using Reserved Pool Capacity might be to guarantee unallocated processor resource to balance workload demand in production workloads.

Attention: On the HMC you can only see the Maximum Pool Capacity and the Reserved Pool Capacity. The Entitled Pool Capacity cannot be displayed on the HMC. To calculate the Entitled Pool Capacity for a Shared-Processor Pool, you have to add the entitlements of all micro-partitions assigned to a Shared-Processor Pool plus the Reserved Pool Capacity.

Although the DB_server and the Apps_server micro-partitions each have 3 virtual processors configured, the total processing capacity which can be used by the two micro-partitions at a time is limited to 3 physical processors. This is due to the Maximum Pool Capacity in the Shared-Processor Pool. If they were assigned to the DefaultPool, they can use up to 6 physical processors if capacity is available.

The NIM_server and the linuxlpar partitions are assigned to the Development pool. The Development pool has a maximum pool capacity of 1 processing unit. The micro-partitions in the Development pool can use at most the capacity of 1 physical processor. The Entitled Pool Capacity of the Development pool is 0.4 as shown in Table 14-2.

Tip: The pool attributes and partition assignments can be changed dynamically while the micro-partitions are running.

Shared-Processor Pool management using the HMC GUI

To configure the pools as just described, perform the following steps:

1. Select the server for which you want to configure the pools. Then select **Configuration** → **Virtual Resources** → **Shared-Processor Pool Management** as shown in Figure 14-1 to open the Shared-Processor Pool configuration window.

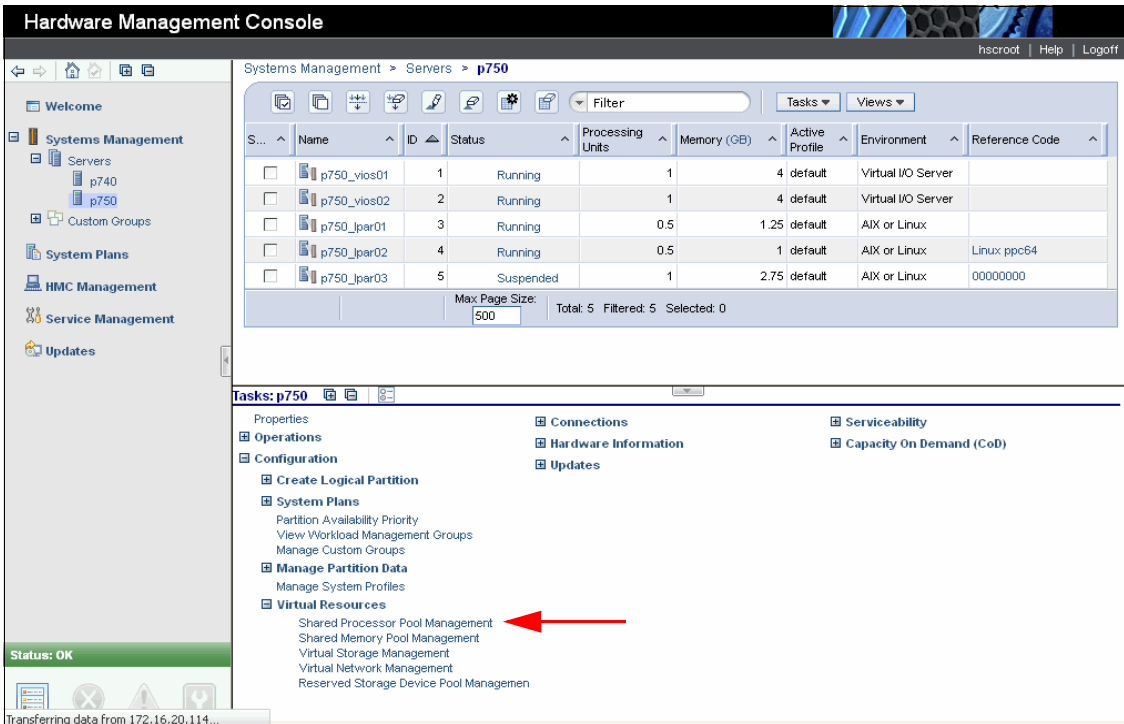


Figure 14-1 Starting Shared-Processor Pool configuration

- Click **SharedPool01**, as shown in Figure 14-2, to open the window for setting the pool attributes.

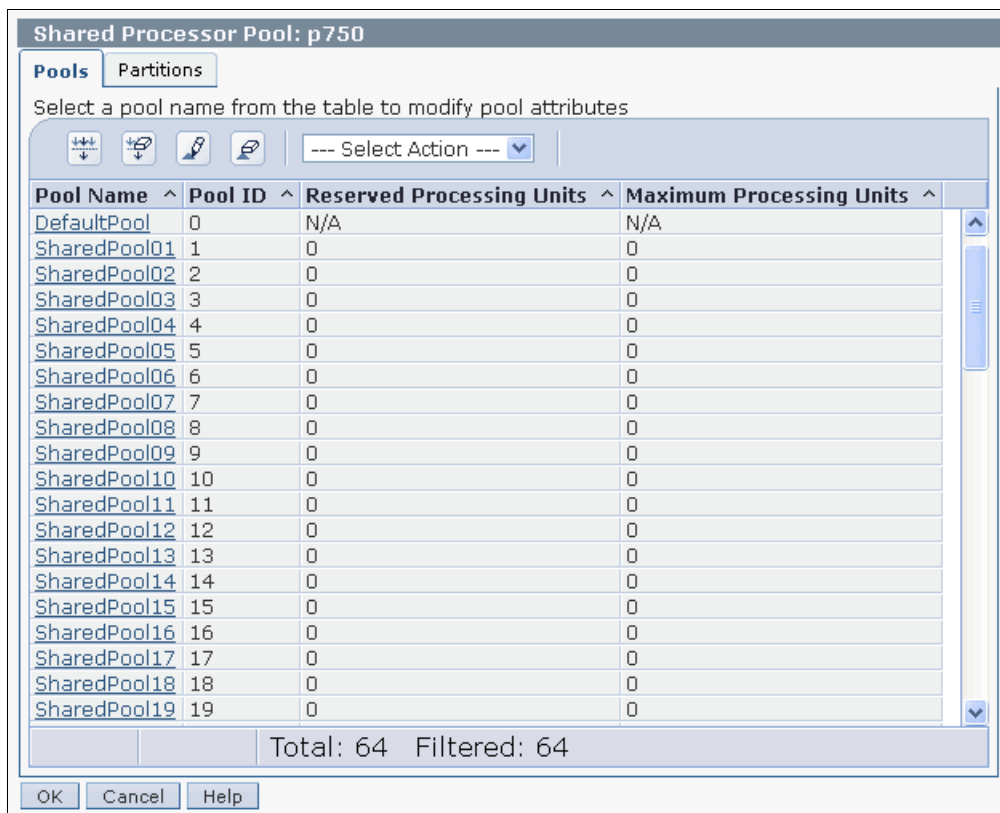
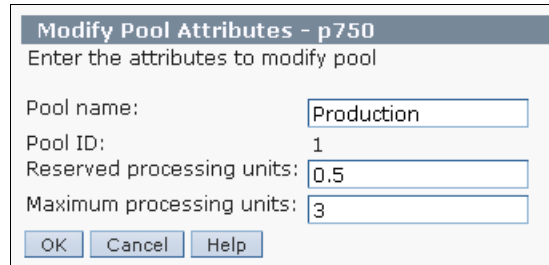


Figure 14-2 Virtual Shared-Processor Pool selection

3. Change the Pool name to Production, Maximum processing units to 3, and Reserved processing units to 0.5, as shown in Figure 14-3. Perform the same for SharedPool02 so that it becomes the Development pool with Maximum processing units of 1.



Modify Pool Attributes - p750
Enter the attributes to modify pool

Pool name:

Pool ID:

Reserved processing units:

Maximum processing units:

Figure 14-3 Shared-Processor Pool configuration

Considerations:

- ▶ The Reserved processing units can be specified in fractions of 1/100 while the Maximum processing units can only be specified in whole numbers.
- ▶ Because the Reserved processing units are guaranteed to the Shared-Processor Pool, the number of Reserved processing units that you specify has to be available on the system.

4. Change to the **Partitions** tab as shown in Figure 14-4 to perform partition assignments. Click each partition that you want to reassign.

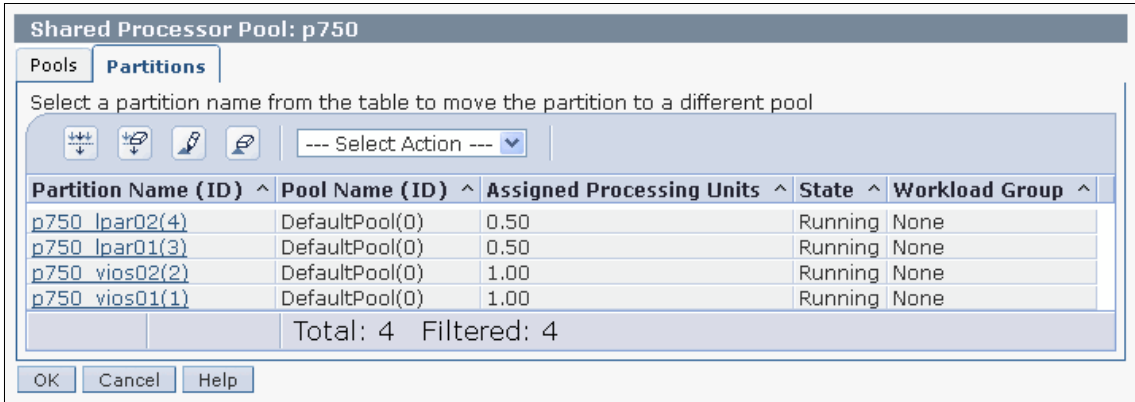


Figure 14-4 Virtual Shared-Processor Pool partition tab

5. Select the desired pool for each partition. Figure 14-5 shows the assignment of the DB_server partition to the production pool. Assign the other partitions according to Table 14-1 on page 389.

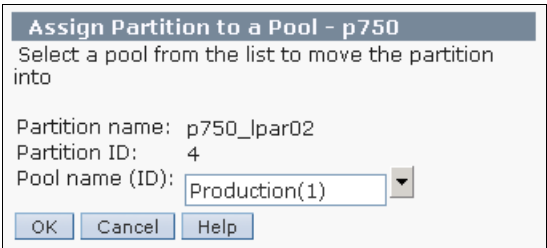


Figure 14-5 Shared-Processor Pool partition assignment

Tip: You cannot assign a micro-partition to a Shared-Processor Pool if it causes the Entitled Pool Capacity to exceed the Maximum Pool Capacity.

Tip: Partitions that have not been activated since they were created do not show up in the window with available partitions like the one shown in Figure 14-4. To change the assignment of a partition that has never been activated to another Shared-Processor Pool, you have to update the Shared-Processor Pool attribute, which can be found in the Processors tab of the partition profile.

Figure 14-6 shows the configuration after all partitions have been assigned.

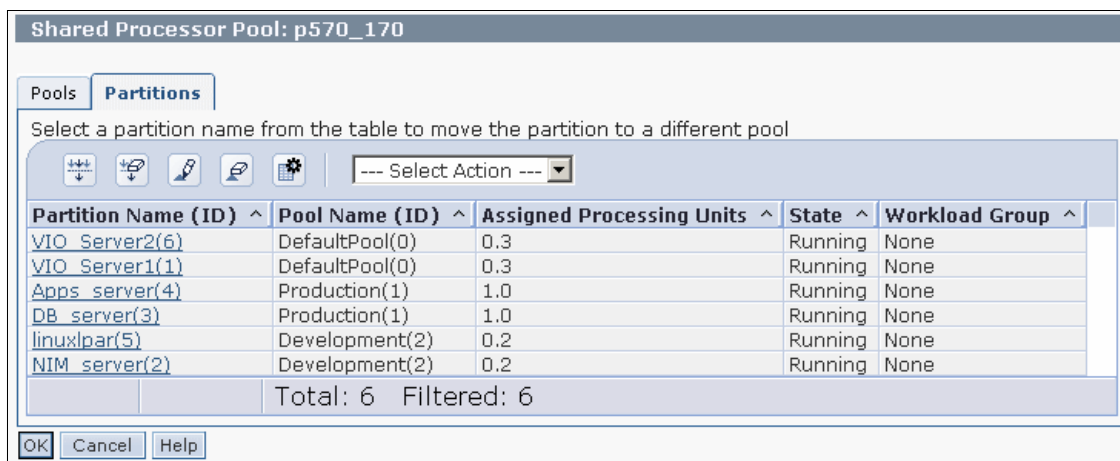


Figure 14-6 Overview of Shared-Processor Pool assignments

Important: If you move a partition to another Shared-Processor Pool, the partition profile will not be updated. Unless you change the profile the partition will be assigned to the original pool the next time it is activated. This is similar to when you are performing dynamic LPAR updates to a partition. When you save the current configuration of a running partition into a new profile, the currently assigned Shared-Processor Pool will be saved.

Shared-Processor Pool management using the command line

You can also manage Shared-Processor Pools through the HMC command line using the **chhwres** command. Example 14-1 shows how to configure the two Shared-Processor Pools as shown in Table 14-2 on page 390.

Example 14-1 Defining Shared-Processor Pools using the command line

```
hscroot@hmc1:~> chhwres -r procpool -m p570_170 -o s --poolid 1 -a
"new_name=Production,max_pool_proc_units=3,reserved_pool_proc_units=0.5"
hscroot@hmc1:~> chhwres -r procpool -m p570_170 -o s --poolid 2 -a
"new_name=Development,max_pool_proc_units=1,reserved_pool_proc_units=0"
```

As shown in Example 14-2, the **lshwres** command can be used to display the Shared-Processor Pool configuration.

Example 14-2 Displaying Shared-Processor Pools using chhwres

```
hscroot@hmc1:~> lshwres -r procpool -m p570_170
name=DefaultPool,shared_proc_pool_id=0,"lpar_names=VIO_Server1,VIO_Server2,DB_server,Apps_server,NIM_server,linuxlpar","lpar_ids=1,6,3,4,2,5"
name=Production,shared_proc_pool_id=1,max_pool_proc_units=3.0,curr_reserved_pool_proc_units=0.0,pend_reserved_pool_proc_units=0.0,lpar_ids=none
name=Development,shared_proc_pool_id=2,max_pool_proc_units=1.0,curr_reserved_pool_proc_units=0.0,pend_reserved_pool_proc_units=0.0,lpar_ids=none
```

The assignment of a partition to a Shared-Processor Pool is also done using the **chhwres** command. Example 14-3 shows the assignment of the example partitions according to Table 14-1 on page 389.

Example 14-3 Assigning partitions to Shared-Processor Pools using chhwres

```
hscroot@hmc1:~> chhwres -r procpool -m p570_170 -o s -p DB_server -a
"shared_proc_pool_name=Production"
hscroot@hmc1:~> chhwres -r procpool -m p570_170 -o s -p Apps_server -a
"shared_proc_pool_name=Production"
hscroot@hmc1:~> chhwres -r procpool -m p570_170 -o s -p NIM_server -a
"shared_proc_pool_name=Development"
hscroot@hmc1:~> chhwres -r procpool -m p570_170 -o s -p linuxlpar -a
"shared_proc_pool_name=Development"
```

Tip: If you try to assign a partition that has never been activated, you get the following error:

This operation is not allowed because the partition is using dedicated processors

To change the assignment of a partition that has never been activated to another Shared-Processor Pool, you have to update the `shared_proc_pool_name` or `shared_proc_pool_id` attributes in the partition profile using the **chsyscfg** command.

Partitions can be dynamically moved between the Shared-Processor Pools. Example 14-4 shows how to move the `linuxlpar` partition from the Development pool to the Production pool.

Example 14-4 Moving a micro-partition to another Shared-Processor Pool

```
hscroot@hmc1:~> chhwres -r procpool -m p570_170 -o s -p linuxlpar -a
"shared_proc_pool_name=Production"
```

Considerations:

- ▶ When you move a partition to another Shared-Processor Pool, the Entitled Pool Capacities change. In our case the Entitled Pool Capacity of the Development pool is reduced by 0.2 while the Entitled Pool Capacity of the Production pool is increased by 0.2.
- ▶ If you move a partition to another Shared-Processor Pool, the partition profile will not be updated. Unless you change the profile, the partition will be assigned to the original pool the next time it is activated. This is similar to when you are performing dynamic LPAR updates to a partition. When you save the current configuration of running partition in to a new profile, the currently assigned Shared-Processor Pool will be saved.

To deactivate a Shared-Processor Pool, all the micro-partitions have to be removed from it and the Maximum Pool Capacity as well as the Reserved Pool Capacity have to be set to 0. Example 14-5 shows the commands required to deactivate the Development pool. The NIM_server micro-partition is moved to the DefaultPool and then the Development pool is set to 0 and renamed to the default name it had initially.

Example 14-5 Deactivating a Shared-Processor Pool using chhwres

```
hscroot@hmc1:~> chhwres -r procpool -m p570_170 -o s -p NIM_server -a  
"shared_proc_pool_name=DefaultPool"  
hscroot@hmc1:~> chhwres -r procpool -m p570_170 -o s --poolid 2 -a  
"new_name=SharedPool02,max_pool_proc_units=0,reserved_pool_proc_units=0"
```

14.2 Shared dedicated capacity

The system administrator can control which dedicated-processor partitions can donate unused cycles. The dedicated-processor partition must be identified as a donating partition.

When the CPU utilization of the core goes below a threshold, and all the SMT threads of the CPU idle from a hypervisor perspective, the CPU will be donated to the Shared-Processor Pool.

The operating system will make a thread idle from hypervisor perspective when it enters the idle loop and the SMT snooze delay expires. The delay needs to be set to zero to maximize the probability of donation. Other than the SMT snooze delay, the donation completes in microseconds.

The following items show how this can be controlled on each operating system:

- ▶ On AIX, the under threshold action is controlled by the `ded_cpu_donate_thresh` schedo tunable. Snooze delay is controlled by the AIX `smt_snooze_delay` and `smt_tertiary_snooze_delay` schedo tunables. Note that the `smt_tertiary_snooze_delay` schedo tunable only applies to POWER7-based and later servers.
- ▶ IBM i supports donation but does not externalize tunable controls. The implementation uses a technique similar to AIX's snooze delay, but with the delay value managed by LIC. With this technique, the extent of donation is dependent on processor utilization as well as the characteristics of the workload, for example, donation will tend to decrease as processor utilization and workload multithreading increase.
- ▶ Linux does not have the concept of `smt_tertiary_snooze_delay` or a direct corollary for `ded_cpu_donate_thresh`. On Linux `smt_snooze_delay` can be set in two different ways:
 - At boot time, it can be set by a kernel command line parameter:
`smt-snooze-delay=100`
The parameter is in microseconds.
 - At runtime, the **ppc64_cpu** command can be used:
`ppc64_cpu --smt-snooze-delay=100`

The donated processor is returned instantaneously (within microseconds) to the dedicated processor partition when one of the following conditions occurs:

- ▶ The timer of one of the SMT threads on the donated CPU expires.
- ▶ An external interrupt for the dedicated-processor partition is presented to one of the SMT threads of the donated CPU.
- ▶ The operating system needs the CPU back to dispatch work on one of the SMT threads of the donated CPU.

LPARs that use processor folding will tend to donate more idle capacity because the workload is constrained to a subset of the available processors and the remaining processors are ceded to the hypervisor on a longer term basis than they are when snooze-delay techniques are used.

A workload in a shared-dedicated partition might see a slight performance impact because of the cache-effects of running micro-partitions on a donated CPU.

Switching a partition from shared-dedicated to dedicated or reverse is a dynamic LPAR operation.

To set up a dedicated donating processor on HMC, do the following steps:

1. Select the partition with dedicated processors.
2. Select **Configuration** → **Manage Profiles** to open the Managed Profiles window.
3. Select a profile and click **Actions** → **Edit** as shown in Figure 14-7.

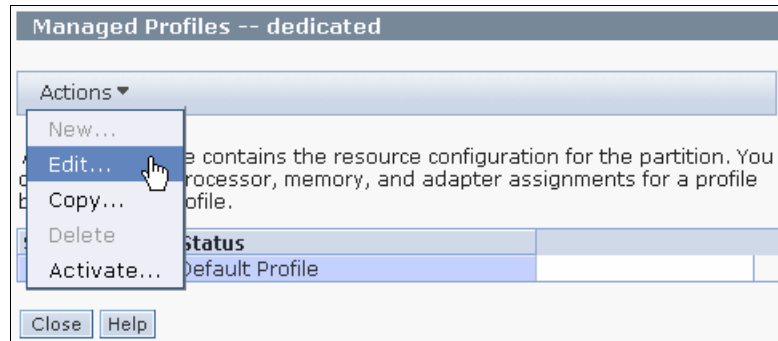


Figure 14-7 The Edit Managed Profile window

4. The Logical Partition Profile Properties is opened. Open the Processors tab as shown in Figure 14-8 where the Processor Sharing options can be set:
 - The option **Allow when partition is inactive** is set by default and indicates whether the dedicated processors are made available to shared-processor partitions when the logical partition that is associated with this partition profile is shut down.
 - The option **Allow when partition is active**, which is available on POWER6 systems or later, indicates whether the dedicated processors are made available to shared-processor partitions when the logical partition that is associated with this partition profile is active.

The screenshot shows the 'Logical Partition Profile Properties' dialog box for 'default @ dedicated @ p570_170 - dedicated'. The 'Processors' tab is selected. The dialog has tabs for General, Processors, Memory, I/O, Virtual Adapters, Power Controlling, Settings, and Logical Host Ethernet Adapters (LHEA). The main content area states: 'Detailed below are the current processing settings for this partition profile.'

Processing mode

- ☒ Dedicated
- ☐ Shared

Dedicated processors

Total managed system processors : 4.00

Minimum processors :

Desired processors :

Maximum processors :

Processor Sharing

- ☒ Allow when partition is inactive.
- ☐ Allow when partition is active.

At the bottom are buttons for OK, Cancel, and Help.

Figure 14-8 Setting the Processor Sharing options

Tip: You can get to the dialog shown in Figure 14-8 while a partition is running by clicking **LPAR properties** to change this dynamically.



Memory virtualization setup

This chapter describes how to configure Active Memory Sharing and how to use Active Memory Deduplication.

It covers the following topics:

- ▶ Active Memory Sharing setup
- ▶ Active Memory Deduplication setup

15.1 Active Memory Sharing setup

This section describes how to configure Active Memory Sharing. The configuration is done using the following three steps:

- ▶ 15.1.1, “Creating the paging devices”
- ▶ 15.1.2, “Creating the shared memory pool” on page 404
- ▶ 15.1.3, “Creating a shared memory partition” on page 411

Note: The examples in the following sections were created using Management Console GUI panels and the Management Console command line interface. The same actions can also be performed using IVM.

15.1.1 Creating the paging devices

A paging device is required for each shared memory partition. The size of the paging device must be equal to or larger than the maximum logical memory defined in the partition profile. The paging devices are owned by a Virtual I/O Server. A paging device can be a logical volume or a whole physical disk. Disks can be local or provided by an external storage subsystem through a SAN.

If you are using whole physical disks, there are no actions required other than making sure that the disks are configured and available on the Virtual I/O Server and any PVID is cleared.

Note: If you plan to use a dual Virtual I/O Server configuration and take advantage of redundancy, your paging devices have to be provided through a SAN and accessible from both Virtual I/O Servers.

Example 15-1 shows the creation of two logical volumes that will be used as paging devices. If you are using logical volumes, you must create a volume group and the logical volumes using the **mkvg** and **mk1v** commands as shown in this example.

Example 15-1 Creating logical volumes as paging devices on the Virtual I/O Server

```
$ mkvg -vg amspaging hdisk2
amspaging
$ mk1v -lv amspaging01 amspaging 5G
amspaging01
$ mk1v -lv amspaging02 amspaging 5G
amspaging02
$ lsvg -lv amspaging
amspaging:
```

LV NAME	TYPE	LPs	PPs	PVs	LV STATE	MOUNT POINT
amspaging01	jfs	40	40	1	closed/syncd	N/A
amspaging02	jfs	40	40	1	closed/syncd	N/A

If you are using IVM to manage your system, this concept, though valid, is more automated. IVM will, upon creating a memory pool, prompt you for a volume group to use for paging. IVM will then automatically create paging devices of the correct size with respect to requirements of your partitions. IVM will also automatically resize the paging device when the partition maximum memory increases, and will delete the device if, through IVM, you delete the partition or switch it to dedicated memory.

If you are configuring SAN storage for paging devices, you should use best practices when assigning SAN LUNs to AMS pools. When possible, use WWPN zoning to ensure that physical paging devices intended for a given AMS pool are only accessible from the one or two Virtual I/O Server partitions that are supporting the AMS pool.

If the paging device is composed of logical volumes on the SAN, each logical volume should be zoned to allow access from only the Virtual I/O Server partition where the logical volume was created. By using zoning to enforce isolation of the paging devices, you will avoid problems where a device is unintentionally assigned for multiple uses simultaneously. When the device is configured for multiple uses, this exposes the contents of the paging devices to possible data integrity problems including corruption.

For information about SAN zoning, see this website:

<http://www.redbooks.ibm.com/abstracts/sg246116.html?Open>

15.1.2 Creating the shared memory pool

This section describes how to create the shared memory pool using the Management Console. In this example we create a shared memory pool with a size of 20 GB and nine paging devices. Paging devices are disks provided through a SAN and mapped on both Virtual I/O Servers. A paging device is required for each shared memory partition. Therefore, the pool will be able to accommodate nine shared memory partitions.

Perform the following steps to create the shared memory pool:

1. On the Management Console, select the managed system on which the shared memory pool will be created. Then select **Configuration** → **Shared Memory Pool Management**, as shown in Figure 15-1.

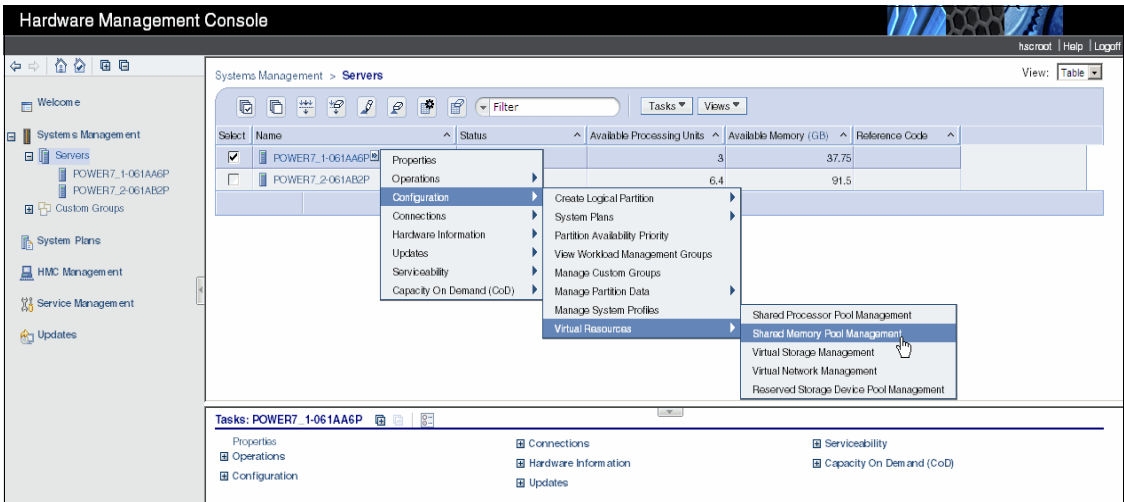


Figure 15-1 Creating a shared memory pool

2. The Welcome window appears, requiring no input. Select **Next** to continue.
3. Enter the Pool size and the Maximum pool size as shown in Figure 15-2 and click **Next**.

The pool size cannot be larger than the amount of available system memory. The memory used by dedicated memory partitions cannot be used for the shared memory pool. In this case, the pool is created on a managed system where we cannot create a pool larger than 62.8 GB.

The Maximum pool size is a soft limit that can be changed dynamically. Keep the Maximum pool size to the required minimum because for each 16 GB of pool memory the hypervisor reserves 256 MB of memory for page tracking. Therefore, if you set the maximum too high, you will be misallocating memory.

Note: Paging space devices can only be assigned to one shared memory pool at a time. You cannot assign the same paging space device to a shared memory pool on one system and to another shared memory pool on another system at the same time.

Figure 15-2 shows how to define the Pool size and the Maximum pool size.

The screenshot displays a web-based configuration interface for creating a shared memory pool. The browser address bar shows the URL <https://hmc9.itso1.ibm.com/> and the page title is "hmc9: Shared Memory Pool Management - Windows ...". The main heading is "Create Shared Memory Pool - POWER7_1-061AA6P".

On the left, a sidebar contains a list of steps: "Welcome" (checked), "General" (highlighted with a yellow arrow), "Paging VIOS", "Paging Space Device(s)", and "Summary".

The "General" tab is active, showing a description: "A shared memory pool defines the amount of shared memory available on the system. Any memory assigned to the pool is not available for use by dedicated memory partitions." Below this, the "Available system memory:" is listed as "62.8 GB".

Two configuration fields are present:

- "Maximum pool size:" with a text input containing "20", a unit selector set to "GB", and a secondary unit selector set to "MB".
- "Pool size:" with a text input containing "20", a unit selector set to "GB", and a secondary unit selector set to "MB".

At the bottom, there is a navigation bar with four buttons: "< Back", "Next >", "Finish", and "Cancel".

Figure 15-2 Defining the Pool size and Maximum pool size

4. As shown in Figure 15-3, select the Virtual I/O Server partitions that will be used from the pull-down menu and click **Next**.

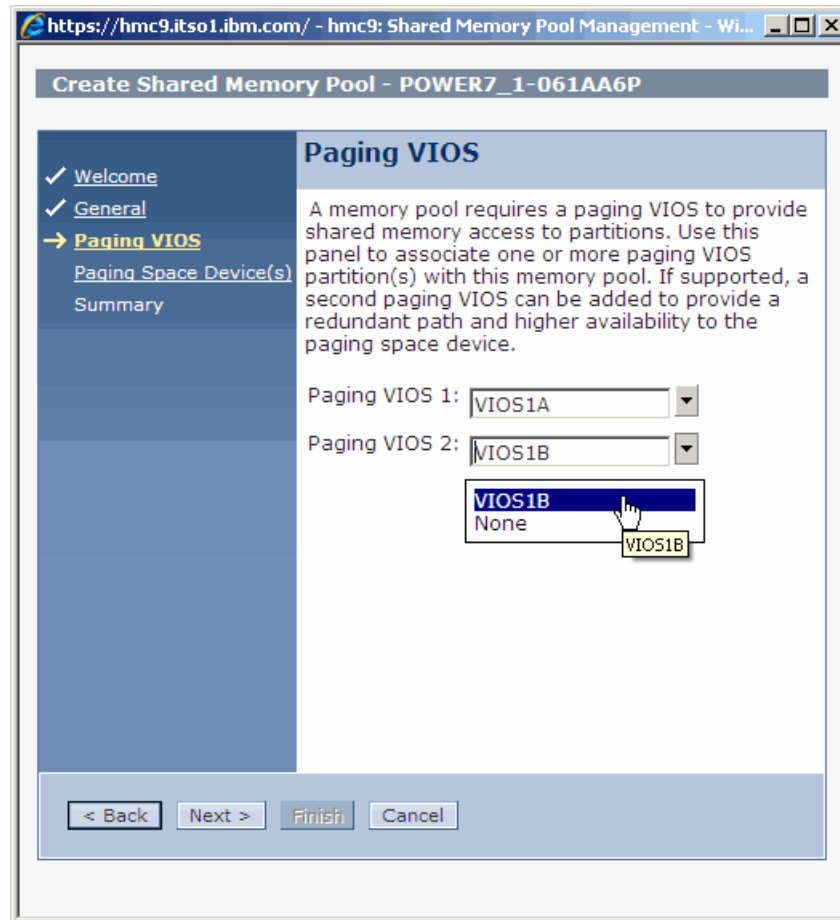


Figure 15-3 Selecting paging space partitions

You can select a maximum of two **Paging Virtual I/O Servers** for redundancy.

5. Click **Select Devices**, as shown in Figure 15-4.

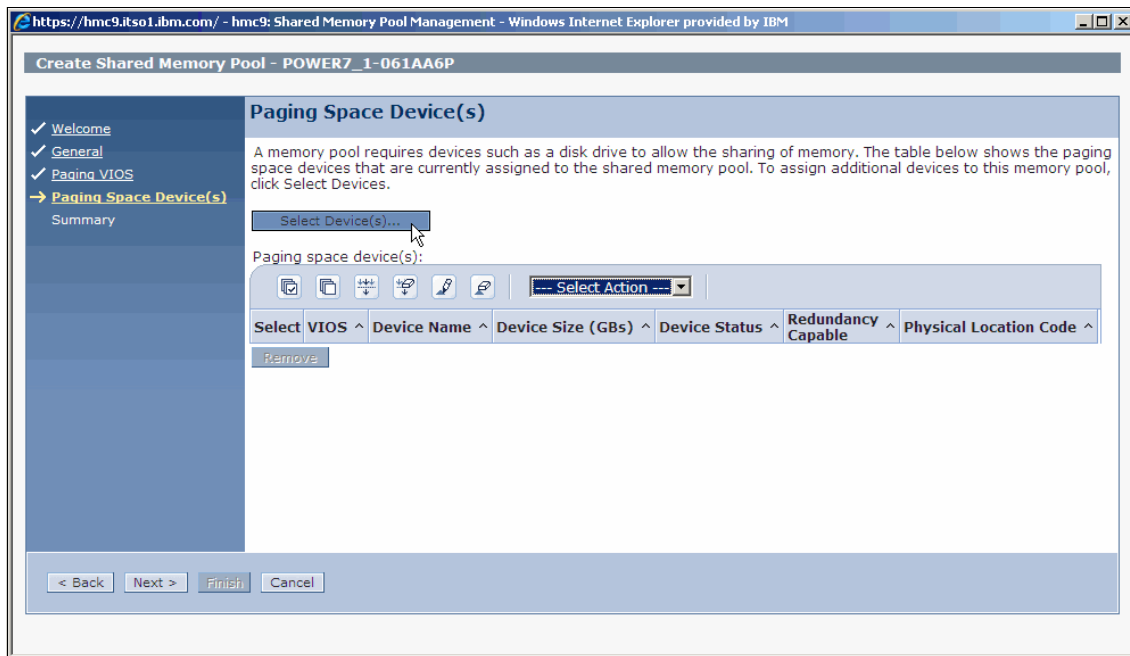


Figure 15-4 Selecting paging devices

6. Click **Refresh**, as shown in Figure 15-5, to display the list of available paging devices. Select the paging devices that should be assigned to the shared memory pool and click **OK**.

In this example, devices with a maximum size of 10 GB are filtered. These devices are displayed in the **Common devices list** because both Virtual I/O Servers can access them. All devices are 10 GB in size. This means that the maximum logical memory setting defined in the partition profile of each of the nine partitions cannot be greater than 10 GB.

Important: The **Redundancy Capable** property only indicates the *capability* for the device to be redundant. It does not mean the device *is* redundant. A redundant device must be seen by both Virtual I/O Servers and therefore is displayed in the **Common devices list**.

Notes:

- ▶ Logical volumes or physical disks that have been mapped to a virtual SCSI adapter will not appear in the selection list.
- ▶ For an IBM i shared memory partition, the paging device must be larger than the maximum memory defined in the partition profile. An IBM i partition requires 1 bit extra for every 16 byte page it allocates. This is a factor of 129/128. The paging size must be greater than or equal to $\text{max mem} * 129/128$. That means an IBM i partition with 10 GB of memory needs a paging device with a minimum size of 10.08 GB of paging space.

Figure 15-5 shows how to select from the available paging devices.

https://hmc9.itso1.ibm.com/#tableTop_2f8a2f8a - hmc9: Shared Memory P...

Paging Space Device Selection - POWER7_1-061AA6P

To display the available paging space devices in the device lists, you must first select filter parameters and then click Refresh. You can list all available paging space devices by selecting All as the device type, or you can narrow your search by selecting a device type, maximum size, or minimum size.

Device Type:

☒ Maximum Size (in GBs):

☐ Minimum Size (in GBs):

Choose from the following list of devices. You can choose more than one paging space device to be added to the pool. Paging space devices should be assigned to only one shared memory pool at a time. You should not assign a paging space device to this shared memory pool if it is already assigned to another shared memory pool on another system.

After you have made your selections, select the OK button to assign the selected devices to the memory pool.

Common device list:

Select	VIOS ^	Device Name ^	Device Size (GBs) ^	Redundancy Capable ^
☒	VIOS1A VIOS1B	hdisk3 hdisk3	10.0	True
☒	VIOS1A VIOS1B	hdisk4 hdisk4	10.0	True
☒	VIOS1A VIOS1B	hdisk5 hdisk5	10.0	True
☒	VIOS1A VIOS1B	hdisk6 hdisk6	10.0	True
☒	VIOS1A VIOS1B	hdisk7 hdisk7	10.0	True
☒	VIOS1A VIOS1B	hdisk8 hdisk8	10.0	True
☒	VIOS1A VIOS1B	hdisk9 hdisk9	10.0	True
☒	VIOS1A VIOS1B	hdisk10 hdisk10	10.0	True
☒	VIOS1A VIOS1B	hdisk11 hdisk11	10.0	True

Figure 15-5 Selecting paging devices

7. In the Summary window, shown in Figure 15-6, click **Finish** to start the creation of the shared memory pool.

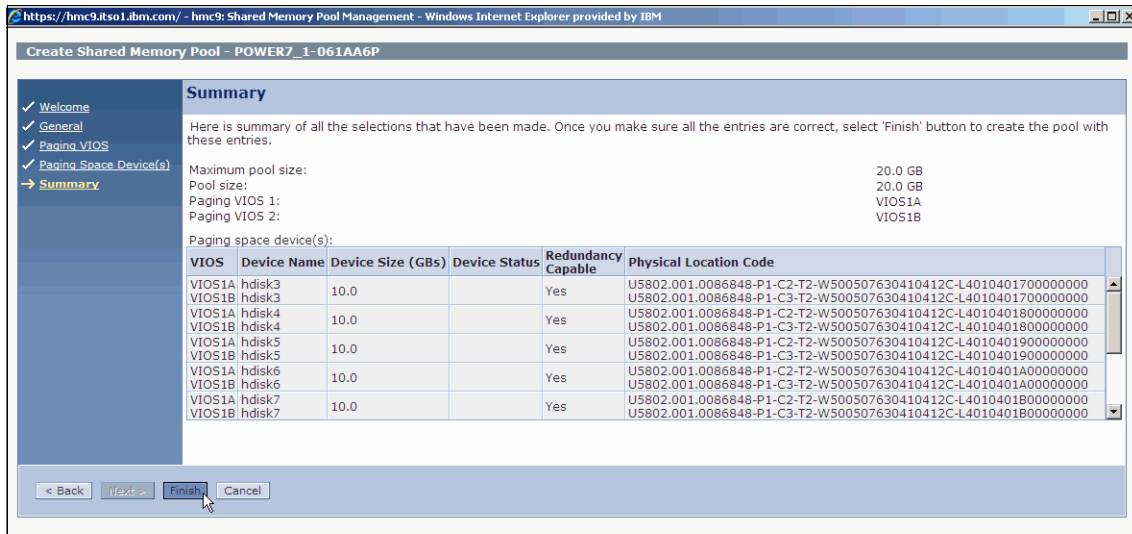


Figure 15-6 Finishing shared memory pool creation

Using the **lshwres** command, the shared memory pool can be displayed on the Management Console command line interface. Using the **-r mempool** flag, as shown in Example 15-2, the attributes of the shared memory pool are displayed.

Example 15-2 Displaying the shared memory pool using lshwres

```
hscroot@hmc9:~> lshwres -m POWER7_1-061AA6P -r mempool
curr_pool_mem=20480,curr_avail_pool_mem=20224,curr_max_pool_mem=20480,p
end_pool_mem=20480,pend_avail_pool_mem=20224,pend_max_pool_mem=20480,sy
s_firmware_pool_mem=256,"paging_vios_names=VIOS1A,VIOS1B","paging_vios_
ids=1,2"
```

When using the **lshwres** command with the **--rsubtype pgdev** flag, as shown in Example 15-3, the attributes and status of each paging device are displayed.

Example 15-3 Displaying paging devices using lshwres

```
hscroot@hmc9:~> lshwres -m POWER7_1-061AA6P -r mempool --rsubtype pgdev
device_name=hdisk3,paging_vios_name=VIOS1A,paging_vios_id=1,size=10240,type=phy
s,state=Inactive,phys_loc=U5802.001.0086848-P1-C2-T1-W500507630410412C-L4010401
700000000,is_redundant=1,redundant_device_name=hdisk3,redundant_paging_vios_nam
e=VIOS1B,redundant_paging_vios_id=2,redundant_state=Inactive,redundant_phys_loc
=U5802.001.0086848-P1-C3-T2-W500507630410412C-L4010401700000000,lpar_id=none
[...]
```

```
device_name=hdisk11,paging_vios_name=VIO1A,paging_vios_id=1,size=10240,type=phys,state=Inactive,phys_loc=U5802.001.0086848-P1-C2-T1-W500507630410412C-L4011401A00000000,is_redundant=1,redundant_device_name=hdisk11,redundant_paging_vios_name=VIO1B,redundant_paging_vios_id=2,redundant_state=Inactive,redundant_phys_loc=U5802.001.0086848-P1-C3-T2-W500507630410412C-L4011401A00000000,lpar_id=none
```

15.1.3 Creating a shared memory partition

The process to create a shared memory partition is similar to creating a partition with dedicated memory. The main difference is in the memory section, where you have to specify that the partition will use shared memory, as shown in Figure 15-7.

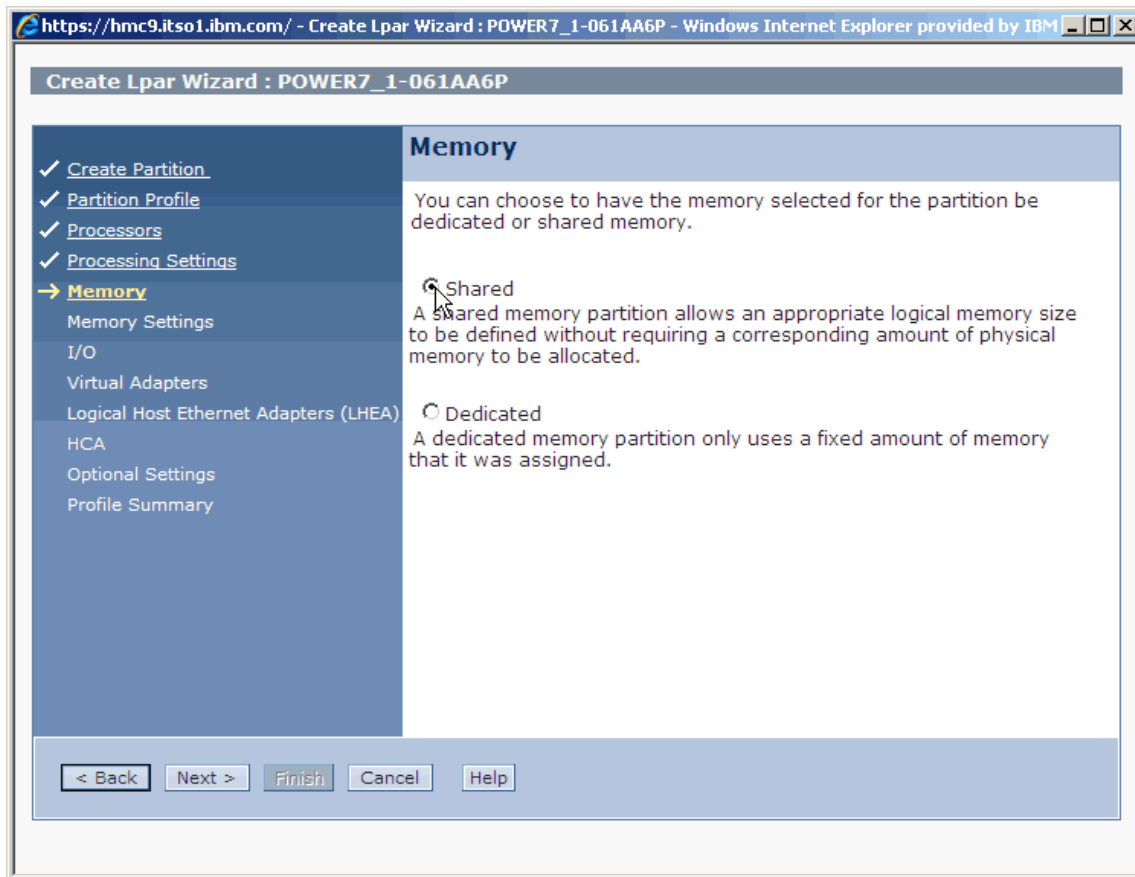


Figure 15-7 Defining a shared memory partition

As shown in Figure 15-8, the memory settings are similar to those of a dedicated memory partition. The **Minimum Memory** setting defines how much logical memory is required to activate the partition. The **Desired Memory** setting is the amount of logical memory that the partition should get. **Maximum Memory** defines the maximum amount of logical memory that can be assigned to the partition using dynamic LPAR.

Notes:

- ▶ In a shared memory partition the Minimum, Desired, and Maximum settings do not represent physical memory values. These settings define the amount of logical memory that is allocated to the partition.
- ▶ The paging device is selected based on the Maximum Memory setting. If there is no paging device available that is larger than or equal to the Maximum Memory, the partition will fail to activate.

Custom Entitled Memory defines how much memory is allowed to be permanently kept in the memory pool to handle I/O activities. It should not be modified unless you have specific requirements regarding I/Os.

You can choose one or two paging VIOS for redundancy. When using two VIOS, VIOS1 is set as the primary paging VIOS for this partition and will therefore be used to handle the paging activity. In case of failure, the hypervisor will switch to the secondary VIOS. Because primary and secondary VIOS are defined at the partition level, you can load balance the paging workload on the two VIOS by defining different paging VIOS on each partition. For partitions using dual VIOS, you must make sure there are redundant paging devices available.

The **Memory Capacity Weight** setting is one of the factors used by the hypervisor to determine which shared memory partition should receive more memory from the shared memory pool. The higher this setting is, the more the partition is likely to get physical memory. However, the main factor for physical memory allocation remains partition activity. This means that a partition with a higher workload will get more physical memory even if it has a lighter weight. This weight factor is mainly useful for situations where two or more partitions have similar activity at the same time and you want to favor one or some of them.

Active Memory Expansion is fully compatible with AMS and therefore allows you to choose an **Active memory expansion factor** if desired.

Figure 15-8 shows how to define the memory settings.

Create Lpar Wizard : POWER7_1-061AA6P

☒ Create Partition
☒ Partition Profile
☒ Processors
☒ Processing Settings
☒ Memory
→ **Memory Settings**
I/O
Virtual Adapters
Logical Host Ethernet Adapters (LHEA)
HCA
Optional Settings
Profile Summary

Memory Settings

Logical Memory

Minimum Memory GB MB

Desired Memory GB MB

Maximum Memory GB MB

Shared Memory Resources

Available pool memory

☐ Custom Entitled Memory MB

VIOS 1

VIOS 2

Memory Capacity Weight

Active Memory Expansion

☐ Active memory expansion factor (1.00 - 10.00)

< Back **Next >** Finish Cancel Help

Figure 15-8 Defining memory settings

Note: By design, the Management Console prevents you from assigning a physical I/O slot or Logical Host Ethernet adapter to a shared memory partition.

15.2 Active Memory Deduplication setup

Active Memory Deduplication acts within an Active Memory Sharing pool, and to use Active Memory Deduplication, you first need to define an Active Memory Sharing pool.

After creating and configuring the Active Memory Sharing pool, you must enable Active Memory Deduplication. You can enable it either from the GUI or CLI of the HMC, as explained in the following sections.

No additional configuration modifications are necessary to allow the LPARs to use Active Memory Deduplication. The LPARs only need to be configured with shared memory to indicate which partitions are needed.

This section includes the following topics:

- ▶ 15.2.1, “Enabling AMD using the HMC GUI”
- ▶ 15.2.2, “Disabling AMD using the HMC GUI” on page 416
- ▶ 15.2.3, “Enabling or disabling AMD using the HMC CLI” on page 417

15.2.1 Enabling AMD using the HMC GUI

To enable Active Memory Deduplication by using the HMC GUI:

1. In the Servers pane of the HMC, check the desired managed system check box, as shown in Figure 15-9.

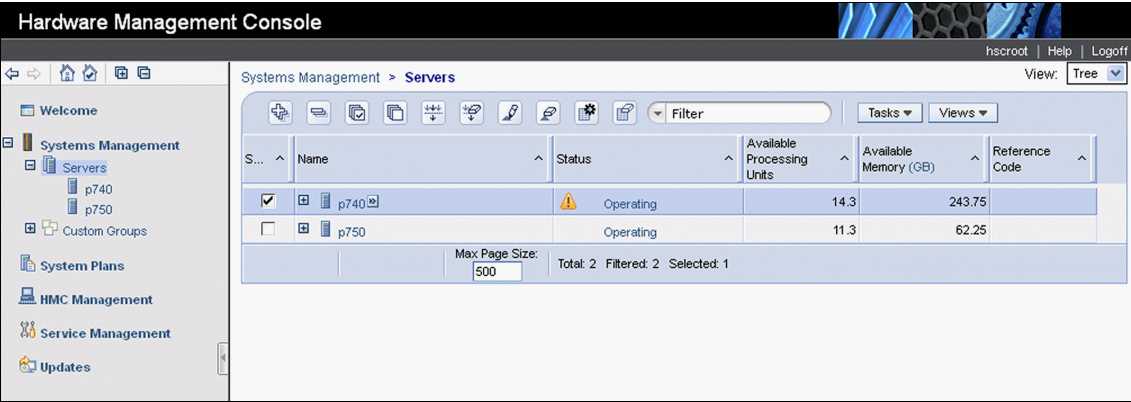


Figure 15-9 Selecting a managed system in the HMC

2. In the Tasks pane of the HMC (Figure 15-10), expand **Configuration** → **Virtual Resources**. Click **Shared Memory Pool Management**.

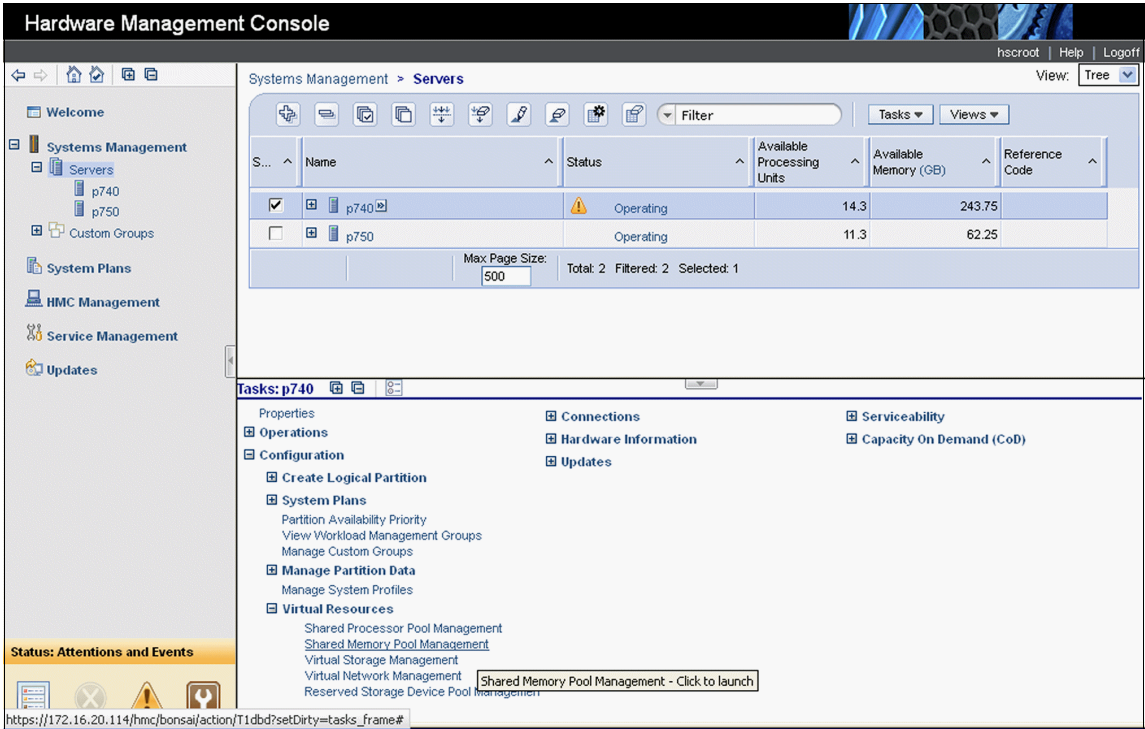


Figure 15-10 Opening the Shared Memory Pool Management window in the HMC

3. In the Pool Properties window (Figure 15-11), check the **Enable Active Memory Deduplication** check box. Click **OK**.

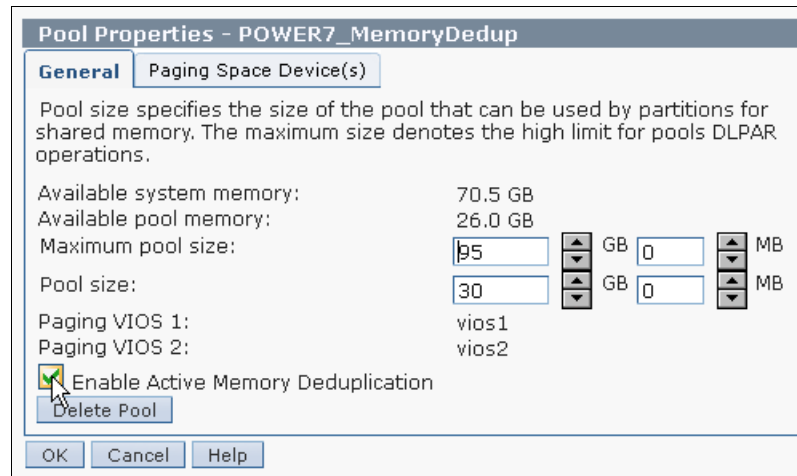


Figure 15-11 Enabling Active Memory Deduplication from HMC Pool Properties

15.2.2 Disabling AMD using the HMC GUI

To disable Active Memory Deduplication, follow the same steps described in 15.2.1, “Enabling AMD using the HMC GUI” on page 414. In the Pool Properties window (Figure 15-12), clear the option **Enable Active Memory Deduplication**, and then click **OK**,

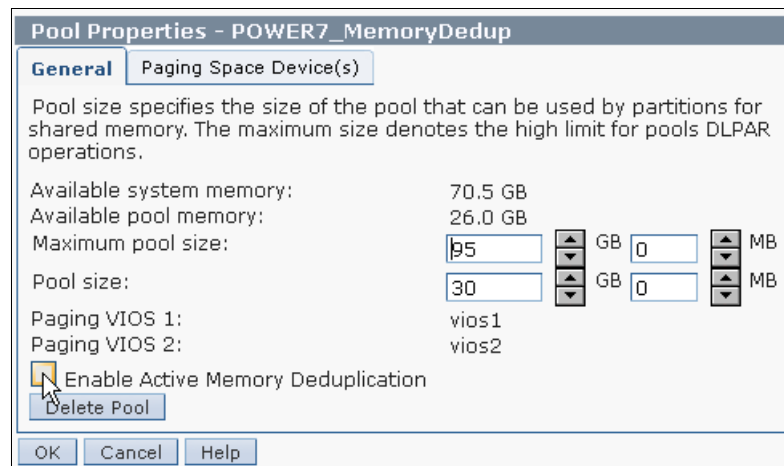


Figure 15-12 Disabling Active Memory Deduplication from HMC Pool Properties

15.2.3 Enabling or disabling AMD using the HMC CLI

To enable Active Memory Deduplication, use the **chhwres** command, as follows:

```
chhwres -r mempool -m <managed system> -o s -a "mem_dedup=1"
```

To disable Active Memory Deduplication, set the **mem_dedup** parameter to 0, as follows:

```
chhwres -r mempool -m <managed system> -o s -a "mem_dedup=0"
```

To check if Active Memory Deduplication is enabled or disabled, use the **lshwres** command, as follows:

```
lshwres -r mempool -m <managed system> -F mem_dedup>
```

If the command returns 1, Active Memory Deduplication is enabled. If it returns a value of 0, Active Memory Deduplication is disabled. Remember to replace the **managed system** field with the name of your managed system, as in the previous examples.



I/O virtualization setup

This chapter introduces the setup of I/O virtualization in a PowerVM environment.

It covers the following topics:

- ▶ Virtual I/O Server setup
- ▶ Storage virtualization setup
- ▶ Network virtualization setup

16.1 Virtual I/O Server setup

This section gives you the details of the operating environment and additional configurations of the Virtual I/O Server. These additional configurations can be done based on the business needs.

16.1.1 Command line interface

The Virtual I/O Server as a virtualization software appliance provides a restricted, scriptable command line interface (IOSCLI). All Virtual I/O Server configurations must be made on this IOSCLI using the restricted shell provided.

Important: Only supported third party storage configuration must be done under the `oem_setup_env` shell environment. The `cfgassist` command offers SMIT-like menus for common configuration tasks, as shown in Figure 16-1.

```
Config Assist for VIOS

Move cursor to desired item and press Enter.

Set Date and TimeZone
Change Passwords
Set System Security
VIOS TCP/IP Configuration
Install and Update Software
Storage Management
Devices
Performance
Role Based Access Control (RBAC)
Shared Storage Pools
Electronic Service Agent

F1=Help      F2=Refresh   F3=Cancel    F8=Image
F9=Shell     F10=Exit     Enter=Do
```

Figure 16-1 Virtual I/O Server Config Assist Menu

The following Virtual I/O Server administration activities are done through the Virtual I/O Server command line interface:

- ▶ Device management (physical and virtual)
- ▶ Network configuration
- ▶ Software installation and update
- ▶ Security
- ▶ User management
- ▶ Installation of OEM software
- ▶ Maintenance tasks

For the initial login to the Virtual I/O Server, use the padmin user ID, which is the primary administrator. Upon login, a password change is required. There is no default.

Upon logging into the Virtual I/O Server, you will be placed into a restricted Korn shell, which works the same way as a regular Korn shell with some restrictions. Specifically, users cannot do the following actions:

- ▶ Change the current working directory.
- ▶ Set the value of the SHELL, ENV, or PATH variable.
- ▶ Specify the path name of the command that contains a redirect output of a command with a >, >|, <>, or >|.

As a result of these restrictions, you are not able to run commands that are not accessible to your PATH variable. These restrictions prevent you from directly sending the output of the command to a file, requiring you to pipe the output to the **tee** command instead.

After you are logged on, you can enter the **help** command to get an overview of the supported commands, as in Example 16-1.

Example 16-1 Supported commands on Virtual I/O Server Version 2.2.2.0

```
vios03:/home/padmin # help
Install Commands
  ioslevel
  license
  lpar_netboot
  lssw
  oem_platform_level
  remote_management
  updateios
LAN Commands
  cfglnagg
Security Commands
  chauth
  chrole
  ldapadd
  ldapsearch
  lsauth
  lsfailedlogin
  lsgcl
  lsrole
  lssecattr
  mkauth
```

cfgnamesrv
 chtcpip
 entstat
 fcstat
 hostmap
 hostname
 lsnetsh
 lstcpip
 mktcpip
 netstat
 optimizenet
 ping
 prepdev
 rmtcpip
 seastat
 startnetsh
 stopnetsh
 traceroute
 vasistat

Device Commands

chdev
 chkdev
 chpath
 cfgdev
 lsdev
 lsmap
 lsnports
 lspath
 mkpath
 mkvdev
 mkvt
 rmdev
 rmpath
 rmvdev
 rmvt
 vfcmap

Physical Volume Commands

lspv
 migratepv

Logical Volume Commands

chlvs
 cplvs

mkkrb5clnt
 mkldap
 mkrole
 rmauth
 rmrole
 rmsecattr
 rolelist
 setkst
 setsecattr
 snmpv3_ssw
 swrole
 tracepriv
 viosecure

UserID Commands

chuser
 lsuser
 mkuser
 passwd
 rmuser

Maintenance Commands

alt_root_vg
 artexdiff
 artexget
 artexlist
 artexmerge
 artexset
 backup
 backupios
 bootlist
 cattracerpt
 cfgassist
 chdate
 chlang
 cl_snmp
 cpvdi
 diagmenu
 dsmd
 errlog
 fsck
 invscout
 ldware
 loginmsg
 lsware

extendlv
lslv
mklv
mklvcopy
rmlv
rmlvcopy

Volume Group Commands

activatevg
chvg
deactivatevg
exportvg
extendvg
importvg
lsvg
mirrorios
mkvg
redefvg
reducevg
syncvg
unmirrorios

Storage Pool Commands

alert
chbdsp
chsp
cluster
lssp
mkbdsp
mksp
rmbdsp
rmsp
snapshot

Virtual Media Commands

chrep
chvopt
loadopt
lsrep
lsvopt
mkrep
mkvopt
rmrep
rmvopt
unloadopt

lslparinfo
motd
mount
pdump
replphyvol
restore
restorevgstruct
save_base
savevgstruct
showmount
shutdown
snap
snmp_info
snmp_trap
startsysdump
starttrace
stoptrace
svmon
sysstat
topas
uname
unmount
viosbr
viostat
vmstat
wkldagent
wkldmgr
wkldout

Monitoring Commands

cfgsvc
lssvc
postprocesssvc
startsvc
stopsvc

Shell Commands

awk
cat
chmod
clear
cp
crontab
date
ftp

```
grep
head
ls
man
mkdir
more
mv
oem_setup_env
rm
sed
stty
tail
tee
vi
wall
wc
who
```

To receive further help about these commands, use the **help** command, as shown in Example 16-2.

Example 16-2 Help command

```
$ help errlog
```

```
Usage: errlog [[ -ls][-seq Sequence_number] | -rm Days]]
```

Displays or clears the error log.

-ls Displays information about errors in the error log file in a detailed format.

-seq Displays information about a specific error in the error log file by the sequence number.

-rm Deletes all entries from the error log older than the number of days specified by the Days parameter.

16.1.2 Mirroring the Virtual I/O Server rootvg

When the installation of the Virtual I/O Server is complete, consider using the following commands to mirror the Virtual I/O Server's rootvg volume group to a second physical volume for redundancy to help protect against Virtual I/O Server outages due to disk failures.

The following steps show how to mirror the Virtual I/O Server rootvg:

1. Use the **extendvg** command to include hdisk2 as part of the rootvg volume group. The same LVM concept applies; you cannot use an hdisk that belongs to another volume group and the disk needs to be of equal size or greater.
2. Use the **lspv** command, as shown in Example 16-3, to confirm that rootvg has been extended to include hdisk2.

Example 16-3 lspv command output before mirroring

```
$ extendvg -f rootvg hdisk2
0516-1162 extendvg: Warning, The Physical Partition Size of 128
requires the creation of 2235 partitions for hdisk2. The limitation
for volume group rootvg is 1016 physical partitions per physical
volume. Use chvg command with -t option to attempt to change the
maximum Physical Partitions per Physical volume for this volume
group.
0516-792 extendvg: Unable to extend volume group.
```

```
$ chvg -factor 6 rootvg
0516-1164 chvg: Volume group rootvg changed. With given
characteristics rootvg can include upto 5 physical volumes with 6096
physical partitions each.
```

```
$ extendvg -f rootvg hdisk2
$
```

```
$ lspv
NAME                PVID                VG
STATUS
hdisk0              00c1f170d7a97dec    rootvg
active
hdisk1              00c1f170e170ae72    rootvg_clients
active
hdisk2              00c1f170e170c9cd    rootvg
hdisk3              00c1f170e170dac6    None
```

3. Use the **mirrorios** command to mirror the rootvg to hdisk1, as shown in Example 16-4. With the -f flag, the **mirrorios** command will automatically reboot the Virtual I/O Server partition.

Attention: SAN disks are usually RAID protected in the storage subsystem. If you use a SAN disk for the rootvg of the Virtual I/O Server, mirroring might not be required.

Example 16-4 Mirroring the Virtual I/O Server rootvg volume group

```
$ mirrorios -f hdisk2
SHUTDOWN PROGRAM
Fri Nov 23 18:35:34 CST 2007
0513-044 The sshd Subsystem was requested to stop.

Wait for 'Rebooting...' before stopping.
```

4. Check if logical volumes are mirrored and if the normal boot sequence has been updated, as shown in Example 16-5. Both mirrored boot devices must appear in the bootlist, when correctly configured.

Example 16-5 Logical partitions are mapped to two physical partitions

```
$ lsvg -lv rootvg
rootvg:
LV NAME          TYPE      LPs      PPs      PVs  LV STATE  MOUNT
POINT
hd5               boot      1        2        2    closed/syncd  N/A
hd6               paging    4        8        2    open/syncd    N/A
paging00          paging    8        16       2    open/syncd    N/A
hd8               jfs2log   1        2        2    open/syncd    N/A
hd4               jfs2      2        4        2    open/syncd    /
hd2               jfs2      23       46       2    open/syncd    /usr
hd9var            jfs2      5        10       2    open/syncd    /var
hd3               jfs2      18       36       2    open/syncd    /tmp
hd1               jfs2      80       160      2    open/syncd    /home
hd10opt           jfs2      6        12       2    open/syncd    /opt
lg_dump1v         sysdump   8        8        1    open/syncd    N/A

$ bootlist -mode normal -ls
hdisk0 blv=hd5
hdisk2 blv=hd5
```

16.1.3 Virtual I/O Server security

This section describes how to harden Virtual I/O Server security using the **viosecure** command. It includes the following topics:

- ▶ Network security
- ▶ The Virtual I/O Server as an LDAP client
- ▶ Network Time Protocol configuration
- ▶ Setting up Kerberos on the Virtual I/O Server
- ▶ Managing users
- ▶ Role-based access control

Network security

If your Virtual I/O Server has an IP address assigned after installation, certain network services are running and open by default. The services in the listening open state are listed in Table 16-1.

Table 16-1 Default open ports on Virtual I/O Server

Port number	Service	Purpose
21	FTP	Unencrypted file transfer
22	SSH	Secure shell and file transfer
23	Telnet	Unencrypted remote login
111	rpcbind	NFS connection
657	RMC	RMC connections (used for dynamic LPAR operations)

In most cases the secure shell (SSH) service for remote login and the secure copy (SCP) for copying files should be sufficient for login and file transfer. Telnet and FTP are not using encrypted communication and can be disabled.

Port 657 for RMC must be left open if you are considering using dynamic LPAR operations. This port is used for the communication between the logical partition and the Hardware Management Console.

Stopping network services

To stop Telnet and FTP and prevent them from starting automatically after reboot, use the **stopnetsvc** command as shown in Example 16-6.

Example 16-6 Stopping network services

```
$ stopnetsvc telnet
0513-127 The telnet subserver was stopped successfully.
$ stopnetsvc ftp
0513-127 The ftp subserver was stopped successfully.
```

Setting up the firewall

The Virtual I/O Server firewall is not enabled by default.

To enable the Virtual I/O Server firewall with the default configuration that enables the services wbem-https, wbem-http, wbem-rmi, rmc, https, http, domain, ssh, ftp and ftp-data, you can use the **viosecure -firewall on -reload** command as shown in Example 16-7.

Example 16-7 Using the viosecure command

```
$ viosecure -firewall on -reload
```

Tip: Default rules are loaded from the /home/ios/security/viosecure.ctl file.

To display the current rules, use the **viosecure -firewall view** command as shown in Example 16-8.

Example 16-8 Displaying the current rules

```
$ viosecure -firewall view
Firewall      ON
```

		ALLOWED		PORTS	
Interface	Local Port	Remote Port	Service	IPAddress	Expiration
Time(seconds)					
-----	----	----	-----	-----	-----
all	5989	any	wbem-https	0.0.0.0	0
all	5988	any	wbem-http	0.0.0.0	0
all	5987	any	wbem-rmi	0.0.0.0	0
all	any	657	rmc	0.0.0.0	0
all	657	any	rmc	0.0.0.0	0
all	443	any	https	0.0.0.0	0
all	any	427	svrloc	0.0.0.0	0
all	427	any	svrloc	0.0.0.0	0
all	80	any	http	0.0.0.0	0
all	any	53	domain	0.0.0.0	0
all	22	any	ssh	0.0.0.0	0
all	21	any	ftp	0.0.0.0	0
all	20	any	ftp-data	0.0.0.0	0

A common approach to designing a firewall or IP filter is to determine ports that are necessary for operation, to determine sources from which those ports will be accessed, and to close everything else. Assume we have hosts on our network as listed in Table 16-2.

Table 16-2 Hosts in the network

Host	IP Address	Comment
VIO Server	172.16.20.171	
Hardware Management Console	172.16.20.111	For dynamic LPAR and for monitoring, RMC communication should be allowed to VIOS.
NIM Server, Management server	172.16.20.41	For administration, SSH communication should be allowed to VIOS.
Administrators workstation	172.16.254.38	SSH communication can be allowed from the administrator's workstation, but it is better use a "jump" to the management server.

Therefore, our firewall would consist of the following rules:

1. Allow RMC from the Hardware Management console.
2. Allow SSH from NIM or the administrator's workstation.
3. Deny anything else.

To deploy this scenario, we issue the **viopsecure -firewall** command to remove all existing default rules and apply new rules.

Tip: You can also set up a firewall from the configuration menu accessed by the **cfgassist** command.

1. Turn off the firewall first so you do not accidentally lock yourself out:
\$ viopsecure -firewall off
2. Remove any existing *allow* rules as shown in Example 16-9.

Example 16-9 Removing the rules

```
$ viopsecure -firewall deny -port 0
The port for the allow rule was not found in the database
```

3. Now set your allow rules. They are going to be inserted before the deny all rule and be matched first. Change the IP addresses used in the example to match your network.

```
$ viosecure -firewall allow -port 657 -address 172.16.20.111
$ viosecure -firewall allow -port 22 -address 172.16.20.41
$ viosecure -firewall allow -port 657 -address 172.16.254.38
```

4. Check your rules. Your output should look like Example 16-10.

Example 16-10 Checking the rules

```
$ viosecure -firewall view
Firewall      OFF
```

		ALLOWED		PORTS	
Interface	Local Port	Remote Port	Service	IPAddress	Expiration
Time(seconds)					
-----	----	----	-----	-----	
-----	----				
all	22	any	ssh	172.16.254.38	0
all	22	any	ssh	172.16.20.41	0
all	657	any	rmc	172.16.20.111	0

5. Turn on your firewall and test connections:

```
$ viosecure -firewall on
```

Important: Lockout can occur if you restrict (that is, if you add a deny rule for) the protocol through which you are connected to the machine.

To avoid lockout, configure the firewall using the virtual terminal connection, *not* the network connection.

Our rule set allows the desired network traffic only and blocks any other requests. The rules set with the **viosecure** command only apply to inbound traffic. However, this setup will also block any ICMP requests, thus making it impossible to ping the Virtual I/O Server or to get any ping responses. This might be an issue if you are using the **ping** command to determine Shared Ethernet Adapter (SEA) failover or for EtherChannel.

Enabling ping through the firewall

As described in “Setting up the firewall” on page 428, our sample firewall setup also blocks all incoming ICMP requests. If you need to enable ICMP for a Shared Ethernet Adapter configuration or Monitoring, use the **oem_setup_env** command and root access to define ICMP rules.

We can create additional ICMP rules that will allow pings by using two commands:

```
/usr/sbin/genfilt -v 4 -a P -s 0.0.0.0 -m 0.0.0.0 -d 0.0.0.0 -M 0.0.0.0 -g  
n -c icmp -o eq -p 0 -0 any -P 0 -r L -w I -l N -t 0 -i all -D echo_reply
```

and:

```
/usr/sbin/genfilt -v 4 -a P -s 0.0.0.0 -m 0.0.0.0 -d 0.0.0.0 -M 0.0.0.0 -g  
n -c icmp -o eq -p 8 -0 any -P 0 -r L -w I -l N -t 0 -i all -D echo_request
```

Security hardening rules

The **viosecure** command can also be used to configure security hardening rules. Users can enforce either the preconfigured security levels or choose to customize them, based on their requirements.

Currently preconfigured rules are high, medium, and low. Each rule has a number of security policies that can be enforced as shown in the following command:

```
$ viosecure -level low -apply  
Processedrules=44      Passedrules=42  Failedrules=2   Level=AllRules  
Input file=/home/ios/security/viosecure.xml
```

Alternatively, users can choose the policies they want as shown in Example 16-11 (the command has been truncated because of its length).

Example 16-11 High level firewall settings

```
$ viosecure -level high
```

1. hls_ISSServerSensorLite:Enable RealSecure Server Sensor Lite: Enables high level policies for RealSecure Server Sensor Lite
2. hls_ISSServerSensorFull:Enable RealSecure Server Sensor Full: Enables high level policies for RealSecure Server Sensor Full
3. hls_tcptr:TCP Traffic Regulation High: Enforces denial-of-service mitigation on popular ports.
4. hls_rootpwdintchk:Root Password Integrity Check: Makes sure that the root password being set is not weak
5. hls_sedconfig:Enable SED feature: Enable Stack Execution Disable feature
6. hls_removeguest:Remove guest account: Removes guest account and its files
7. hls_chetcftusers:Add root user in /etc/ftpusers file: Adds root username in /etc/ftpusers file
8. hls_xhost:Disable X-Server access: Disable access control for X-Server
9. hls_rmdotfrmpathnroot:Remove dot from non-root path: Removes dot from PATH environment variable from files .profile, .kshrc, .cshrc and .login in user's home directory
10. hls_rmdotfrmpathroot:Remove dot from path root: Remove dot from PATH environment variable from files .profile, .kshrc, .cshrc and .login in root's home directory

```
? 1,2
```

11. hls_loginherald:Set login herald: Set login herald in default stanza
12. hls_crontabperm:Crontab permissions: Ensures root's crontab jobs are owned and writable only by root
13. hls_limitsysacc:Limit system access: Makes root the only user in cron.allow file and removes the cron.deny file
14. hls_core:Set core file size: Specifies the core file size to 0 for root
15. hls_umask:Object creation permissions: Specifies default object creation permissions to 077
16. hls_ipsecshunports:Guard host against port scans: Shuns vulnerable ports for 5 minutes to guard the host against port scans
17. hls_ipsecshunhost:Shun host for 5 minutes: Shuns the hosts for 5 minutes, which tries to access un-used ports
18. hls_sockthresh:Network option sockthresh: Set network option sockthresh's value to 60

To view the current security rules, use the **viosecure -view** command.

To undo all security policies, use the **viosecure -undo** command.

DoS hardening

To overcome Denial of Service attacks, a feature was implemented in a Virtual I/O Server. For more information about this topic “Denial of Service hardening” on page 615.

The Virtual I/O Server as an LDAP client

The Lightweight Directory Access Protocol defines a standard method for accessing and updating information about a directory (a database) either locally or remotely in a client-server model. The LDAP method is used by a cluster of hosts to allow centralized security authentication and access to user and group information.

Virtual I/O Server Version 1.4 introduced LDAP authentication for the Virtual I/O Server's users and with Version 1.5 of Virtual I/O Server a secure LDAP authentication is also supported, using a secure sockets layer (SSL). LDAP is packaged on the Virtual I/O Server Expansion Pack media.

The steps necessary to create an SSL certificate, set up a server and then configure the Virtual I/O Server as a client are described in the following sections.

Creating a key database file

All the steps described here suppose that an IBM Tivoli Directory Server is installed on one server in the environment and the GSKit file sets. More information about the IBM Tivoli Directory Server can be found at:

<http://www.ibm.com/software/tivoli/products/directory-server/>

To create the key database file and certificate (self-signed for simplicity in this example), follow these steps:

1. Ensure that the GSKit and gsk7ikm are installed on the LDAP server as follows:

```
# ls|pp -l |grep gsk
gskjs.rte          7.0.3.30  COMMITTED  AIX Certificate and SSL Java
gksa.rte           7.0.3.30  COMMITTED  AIX Certificate and SSL Base
gskta.rte           7.0.3.30  COMMITTED  AIX Certificate and SSL Base
```


2. Start the gsk7ikm utility with X Window. This is located in /usr/bin/gsk7ikm, which is a symbolic link to /usr/opt/ibm/gskta/bin/gsk7ikm. A window like the one shown in Figure 16-2 will appear.

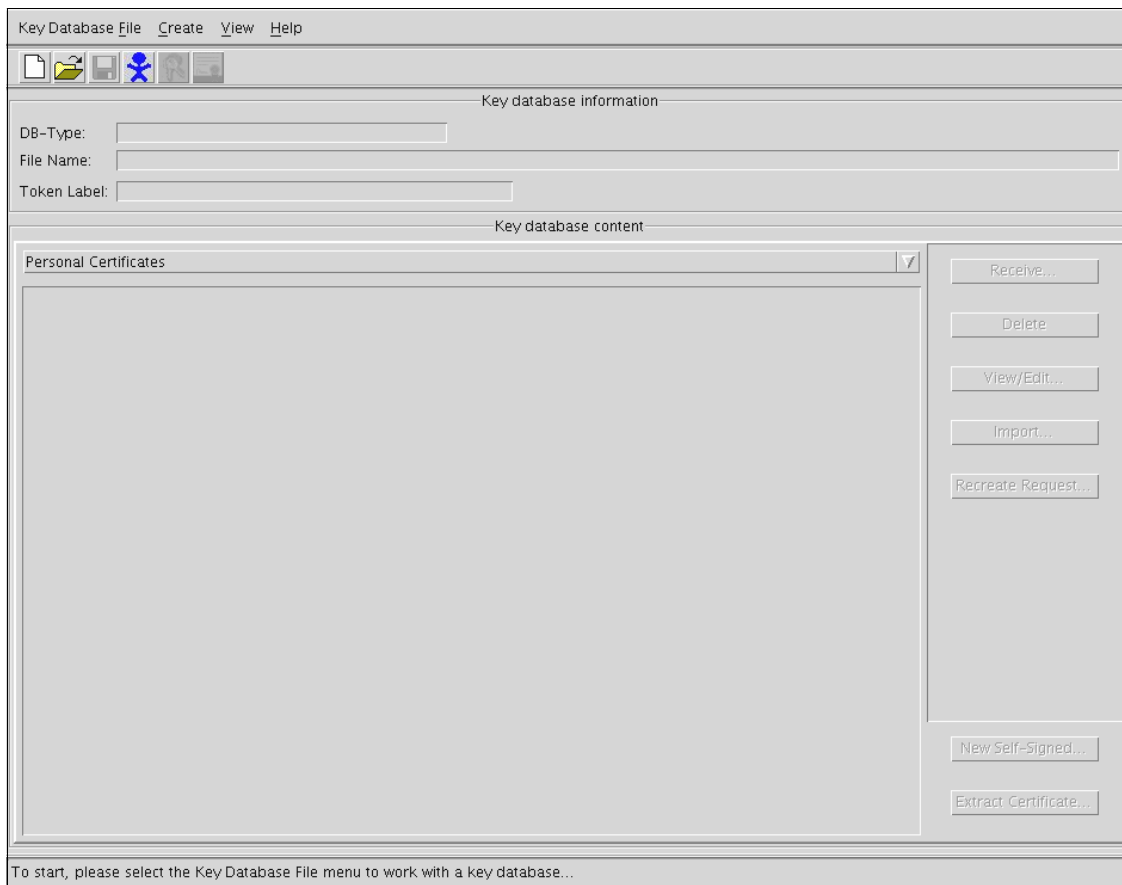
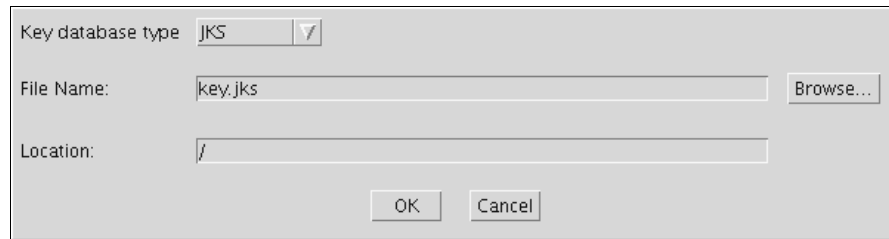


Figure 16-2 The ikeyman program initial window

3. Click **Key Database File** → **New**. A window similar to the one in Figure 16-3 will appear.



Key database type: JKS

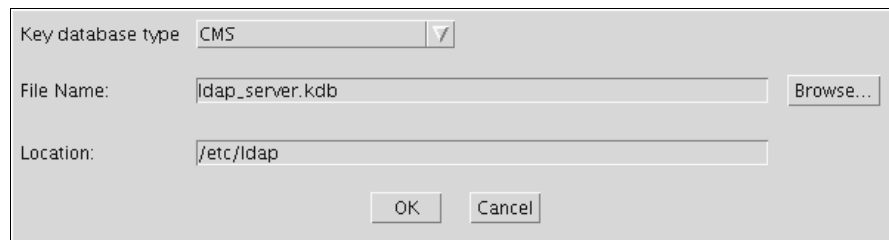
File Name: key.jks Browse...

Location: /

OK Cancel

Figure 16-3 Create new key database window

4. On the same window, change the Key database type to CMS, change the File Name (to `ldap_server.kdb`, in this example), and set the Location to a directory where the keys can be stored (`/etc/ldap`, in this example). The final window will be similar to Figure 16-4.



Key database type: CMS

File Name: ldap_server.kdb Browse...

Location: /etc/ldap

OK Cancel

Figure 16-4 Creating the ldap_server key

5. Click **OK**.
6. A new window will appear. Enter the key database file password, and confirm it. Remember this password because it is required when the database file is edited. In this example the key database password was set to `passw0rd`.
7. Accept the default expiration time.

8. If you want the password to be masked and stored in a stash file, select **Stash the password to a file**.

A stash file can be used by certain applications so that the application does not have to know the password to use the key database file. The stash file has the same location and name as the key database file and has an extension of *.sth.

The panel should be similar to the one shown in Figure 16-5.



Password: *****

Confirm Password: *****

☐ Set expiration time? 60 Days

☒ Stash the password to a file?

Password Strength:

Five key icons representing password strength: three are active (shiny) and two are disabled (gray).

OK Reset Cancel

Figure 16-5 Setting the key database password

9. Click **OK**.

This completes the creation of the key database file. There is a set of default signer certificates. These are the default certificate authorities that are recognized. This is shown in Figure 16-6.

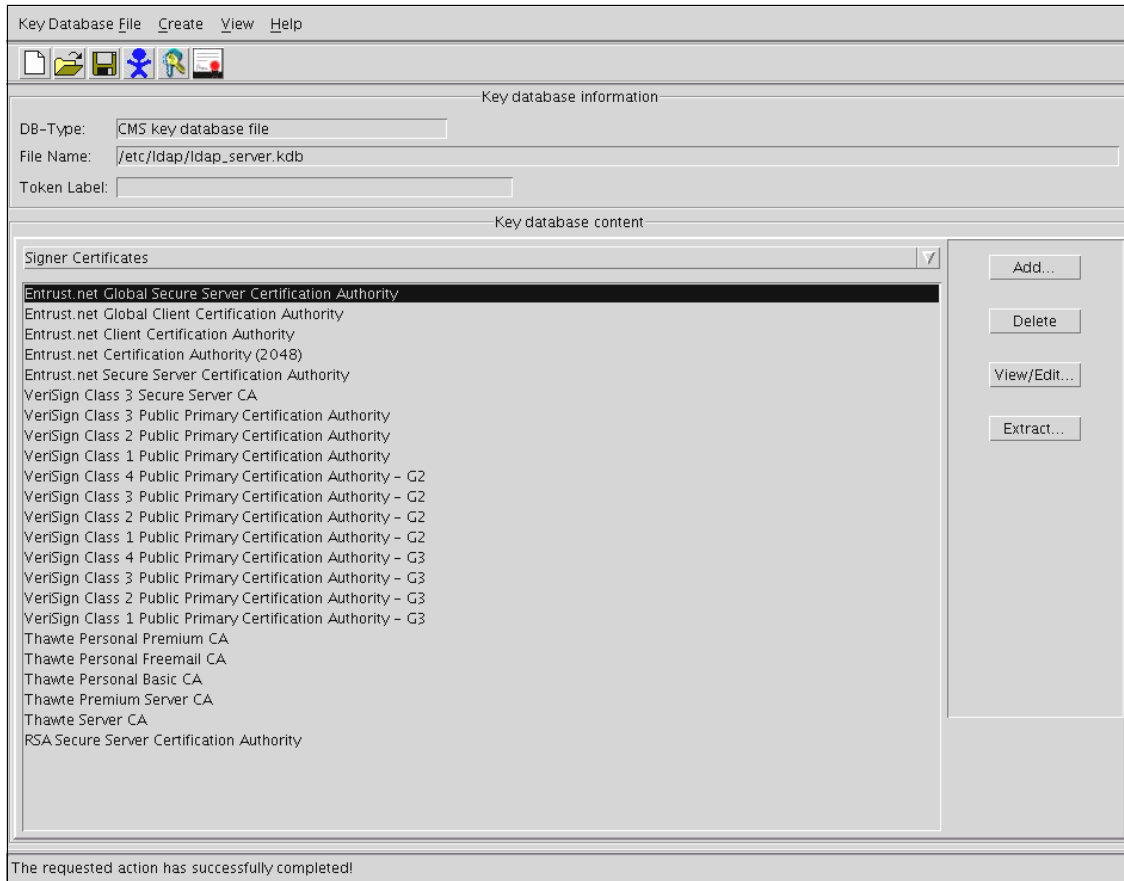
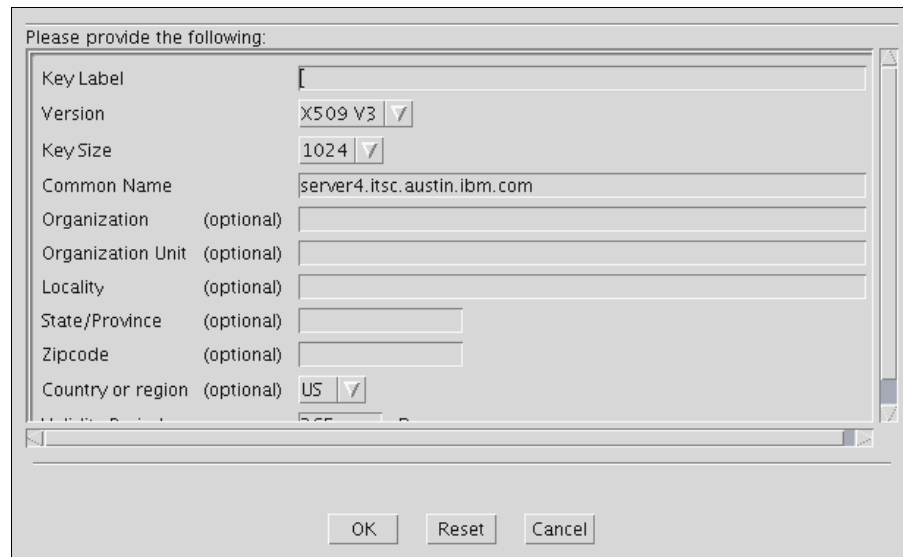


Figure 16-6 Default certificate authorities available on the ikeyman program

10. At this time, the key could be exported and sent to a certificate authority to be validated and then used. In this example, for simplicity reasons, the key is signed using a self-signed certificate. To create a self-signed certificate, click **Create** → **New Self-Signed Certificate**. A window similar to the one in Figure 16-7 will appear.



Please provide the following:

Key Label	
Version	X509 V3
Key Size	1024
Common Name	server4.itsc.austin.ibm.com
Organization (optional)	
Organization Unit (optional)	
Locality (optional)	
State/Province (optional)	
Zipcode (optional)	
Country or region (optional)	US

OK Reset Cancel

Figure 16-7 Creating a self-signed certificate initial panel

11. Type a name in the Key Label field that GSKit can use to identify this new certificate in the key database. In this example the key is labeled `ldap_server`.
12. Accept the defaults for the Version field (X509V3) and for the Key Size field.
13. Type your company name in the Organization field.

14. Complete any optional fields or leave them blank: the default for the Country field and 365 for the Validity Period field. The window should look like the one in Figure 16-8.

Please provide the following:

Key Label		ldap_server
Version		X509 V3 ▾
Key Size		1024 ▾
Common Name		ldap_server.itsc.austin.ibm.com
Organization	(optional)	IBM
Organization Unit	(optional)	ITSO
Locality	(optional)	Austin
State/Province	(optional)	Texas
Zipcode	(optional)	
Country or region	(optional)	US ▾

OK Reset Cancel

Figure 16-8 Self-signed certificate information

15. Click **OK**. GSKit generates a new public and private key pair and creates the certificate.

This completes the creation of the LDAP client's personal certificate. It is displayed in the Personal Certificates section of the key database file.

Next, the LDAP Server's certificate must be extracted to a Base64-encoded ASCII data file.

16. Highlight the self-signed certificate that was just created.
17. Click **Extract Certificate**.
18. Select **Base64-encoded ASCII data** as the type.
19. Type a certificate file name for the newly extracted certificate. The certificate file's extension is usually *.arm.
20. Type the location where you want to store the extracted certificate and then click **OK**.
21. Copy this extracted certificate to the LDAP server system.

This file will only be used if the key database is going to be used as an SSL in a web server. This can happen when the LDAP administrator decides to manage the LDAP through its web interface. Then this *.arm file can be transferred to your PC and imported to the web browser.

You can find more about the GSKit at:

http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.itame.doc_5.1/am51_webinstall223.htm

Configuring the LDAP server

Because the key database was generated, it can now be used to configure the LDAP server.

In the following example, we use a LDAP server on AIX. IBM i and Linux can also be used for LDAP server instead of AIX, depending on your situation. For further information about IBM i LDAP server support, see the IBM i Information Center at:

<http://publib.boulder.ibm.com/infocenter/iserics/v7r1m0/index.jsp>

Navigate to **IBM i 7.1 Information Center → Networking → TCP/IP applications, protocols, and services → IBM Tivoli Directory Server for IBM i (LDAP)**. For Linux, see the relevant product documentation.

The **mksecldap** command is used to set up an AIX system as an LDAP server or client for security authentication and data management.

A description of how to set up the AIX system as an LDAP server is provided in this section. Remember that all file sets of the IBM Tivoli directory Server 6.1 have to be installed before configuring the system as an LDAP server. When installing the LDAP server file set, the LDAP client file set and the backend DB2 software are automatically installed as well. No DB2 preconfiguration is required to run this command for the LDAP server setup. When the **mksecldap** command is run to set up a server, the command does the following:

1. Creates the DB2 instance with ldapdb2 as the default instance name.
2. Because in this case the IBM Directory Server 6.1 is being configured, an LDAP server instance with the default name of ldapdb2 is created. A prompt is displayed for the encryption seed to create the key files. The input encryption seed must be at least 12 characters.
3. Creates a DB2 database with ldapdb2 as the default database name.

4. Creates the base DN (o=ibm in this example). The directory information tree that will be created in this example by default is shown in Figure 16-9.

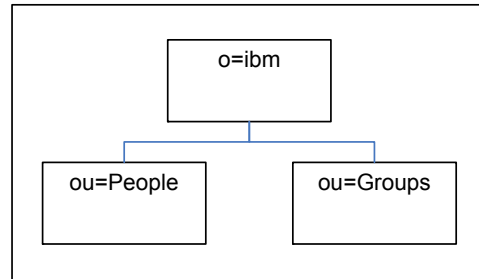


Figure 16-9 Default directory information tree created by mksecdap command

1. Because the **-u NONE** flag was not specified, the data from the security database from the local host is exported into the LDAP database. Because the **-S** option was used and followed by `rfc2307aix`, the **mksecdap** command exports users or groups using this schema.
2. The LDAP administrator DN is set to `cn=admin` and the password is set to `passw0rd`.
3. Because the **-k** flag was used, the server will use SSL (secure socket layer).
4. The plugin `libldapaudit.a` is installed. This plugin supports an AIX audit of the LDAP server.
5. The LDAP server is started after all the above steps are completed.
6. The LDAP process is added to `/etc/inittab` to have the LDAP server start after a reboot.

The command and its output are shown here:

```
# mksecdap -s -a cn=admin -p passw0rd -S rfc2307aix -d o=ibm -k /etc/ldap/ldap_server.kdb -w
passw0rd
ldapdb2's New password:
Enter the new password again:
Enter an encryption seed to generate key stash files:
You have chosen to perform the following actions:

GLPICR020I A new directory server instance 'ldapdb2' will be created.
GLPICR057I The directory server instance will be created at: '/home/ldapdb2'.
GLPICR013I The directory server instance's port will be set to '389'.
GLPICR014I The directory server instance's secure port will be set to '636'.
GLPICR015I The directory instance's administration server port will be set to '3538'.
GLPICR016I The directory instance's administration server secure port will be set to '3539'.
GLPICR019I The description will be set to: 'IBM Tivoli Directory Server Instance V6.1'.
```



```

GLPICR021I Database instance 'ldapdb2' will be configured.
GLPICR028I Creating directory server instance: 'ldapdb2'.
GLPICR025I Registering directory server instance: 'ldapdb2'.
GLPICR026I Registered directory server instance: : 'ldapdb2'.
GLPICR049I Creating directories for directory server instance: 'ldapdb2'.
GLPICR050I Created directories for directory server instance: 'ldapdb2'.
GLPICR043I Creating key stash files for directory server instance: 'ldapdb2'.
GLPICR044I Created key stash files for directory server instance: 'ldapdb2'.
GLPICR040I Creating configuration file for directory server instance: 'ldapdb2'.
GLPICR041I Created configuration file for directory server instance: 'ldapdb2'.
GLPICR034I Creating schema files for directory server instance: 'ldapdb2'.
GLPICR035I Created schema files for directory server instance: 'ldapdb2'.
GLPICR037I Creating log files for directory server instance: 'ldapdb2'.
GLPICR038I Created log files for directory server instance: 'ldapdb2'.
GLPICR088I Configuring log files for directory server instance: 'ldapdb2'.
GLPICR089I Configured log files for directory server instance: 'ldapdb2'.
GLPICR085I Configuring schema files for directory server instance: 'ldapdb2'.
GLPICR086I Configured schema files for directory server instance: 'ldapdb2'.
GLPICR073I Configuring ports and IP addresses for directory server instance: 'ldapdb2'.
GLPICR074I Configured ports and IP addresses for directory server instance: 'ldapdb2'.
GLPICR077I Configuring key stash files for directory server instance: 'ldapdb2'.
GLPICR078I Configured key stash files for directory server instance: 'ldapdb2'.
GLPICR046I Creating profile scripts for directory server instance: 'ldapdb2'.
GLPICR047I Created profile scripts for directory server instance: 'ldapdb2'.
GLPICR069I Adding entry to /etc/inittab for the administration server for directory instance:
'ldapdb2'.
GLPICR070I Added entry to /etc/inittab for the administration server for directory instance:
'ldapdb2'.
GLPICR118I Creating runtime executable for directory server instance: 'ldapdb2'.
GLPICR119I Created runtime executable for directory server instance: 'ldapdb2'.
GLPCTL074I Starting admin daemon instance: 'ldapdb2'.
GLPCTL075I Started admin daemon instance: 'ldapdb2'.
GLPICR029I Created directory server instance: : 'ldapdb2'.
GLPICR031I Adding database instance 'ldapdb2' to directory server instance: 'ldapdb2'.
GLPCTL002I Creating database instance: 'ldapdb2'.
GLPCTL003I Created database instance: 'ldapdb2'.
GLPCTL017I Cataloging database instance node: 'ldapdb2'.
GLPCTL018I Cataloged database instance node: 'ldapdb2'.
GLPCTL008I Starting database manager for database instance: 'ldapdb2'.
GLPCTL009I Started database manager for database instance: 'ldapdb2'.
GLPCTL049I Adding TCP/IP services to database instance: 'ldapdb2'.
GLPCTL050I Added TCP/IP services to database instance: 'ldapdb2'.
GLPICR081I Configuring database instance 'ldapdb2' for directory server instance: 'ldapdb2'.
GLPICR082I Configured database instance 'ldapdb2' for directory server instance: 'ldapdb2'.
GLPICR052I Creating DB2 instance link for directory server instance: 'ldapdb2'.
GLPICR053I Created DB2 instance link for directory server instance: 'ldapdb2'.
GLPICR032I Added database instance 'ldapdb2' to directory server instance: 'ldapdb2'.
You have chosen to perform the following actions:

```

GLPDPW004I The directory server administrator DN will be set.
GLPDPW005I The directory server administrator password will be set.
GLPDPW009I Setting the directory server administrator DN.
GLPDPW010I Directory server administrator DN was set.
GLPDPW006I Setting the directory server administrator password.
GLPDPW007I Directory server administrator password was set.
You have chosen to perform the following actions:

GLPCDB023I Database 'ldapdb2' will be configured.
GLPCDB024I Database 'ldapdb2' will be created at '/home/ldapdb2'
GLPCDB035I Adding database 'ldapdb2' to directory server instance: 'ldapdb2'.
GLPCTL017I Cataloging database instance node: 'ldapdb2'.
GLPCTL018I Cataloged database instance node: 'ldapdb2'.
GLPCTL008I Starting database manager for database instance: 'ldapdb2'.
GLPCTL009I Started database manager for database instance: 'ldapdb2'.
GLPCTL026I Creating database: 'ldapdb2'.
GLPCTL027I Created database: 'ldapdb2'.
GLPCTL034I Updating the database: 'ldapdb2'
GLPCTL035I Updated the database: 'ldapdb2'
GLPCTL020I Updating the database manager: 'ldapdb2'.
GLPCTL021I Updated the database manager: 'ldapdb2'.
GLPCTL023I Enabling multi-page file allocation: 'ldapdb2'
GLPCTL024I Enabled multi-page file allocation: 'ldapdb2'
GLPCDB005I Configuring database 'ldapdb2' for directory server instance: 'ldapdb2'.
GLPCDB006I Configured database 'ldapdb2' for directory server instance: 'ldapdb2'.
GLPCTL037I Adding local loopback to database: 'ldapdb2'.
GLPCTL038I Added local loopback to database: 'ldapdb2'.
GLPCTL011I Stopping database manager for the database instance: 'ldapdb2'.
GLPCTL012I Stopped database manager for the database instance: 'ldapdb2'.
GLPCTL008I Starting database manager for database instance: 'ldapdb2'.
GLPCTL009I Started database manager for database instance: 'ldapdb2'.
GLPCDB003I Added database 'ldapdb2' to directory server instance: 'ldapdb2'.
You have chosen to perform the following actions:

GLPCSF007I Suffix 'o=ibm' will be added to the configuration file of the directory server instance 'ldapdb2'.
GLPCSF004I Adding suffix: 'o=ibm'.
GLPCSF005I Added suffix: 'o=ibm'.
GLPSRV034I Server starting in configuration only mode.
GLPCOM024I The extended Operation plugin is successfully loaded from libevent.a.
GLPSRV155I The DIGEST-MD5 SASL Bind mechanism is enabled in the configuration file.
GLPCOM021I The preoperation plugin is successfully loaded from libDigest.a.
GLPCOM024I The extended Operation plugin is successfully loaded from libevent.a.
GLPCOM024I The extended Operation plugin is successfully loaded from libtrnnext.a.
GLPCOM023I The postoperation plugin is successfully loaded from libpsearch.a.
GLPCOM024I The extended Operation plugin is successfully loaded from libpsearch.a.
GLPCOM025I The audit plugin is successfully loaded from libldapaudit.a.
GLPCOM024I The extended Operation plugin is successfully loaded from libevent.a.
GLPCOM023I The postoperation plugin is successfully loaded from libpsearch.a.

GLPCOM024I The extended Operation plugin is successfully loaded from libpsearch.a.
 GLPCOM022I The database plugin is successfully loaded from libback-config.a.
 GLPCOM024I The extended Operation plugin is successfully loaded from libloga.a.
 GLPCOM024I The extended Operation plugin is successfully loaded from libidsfget.a.
 GLPSRV180I Pass-through authentication is disabled.
 GLPCOM003I Non-SSL port initialized to 389.
 Stopping the LDAP server.
 GLPSRV176I Terminated directory server instance 'ldapdb2' normally.
 GLPSRV041I Server starting.
 GLPCTL113I Largest core file size creation limit for the process (in bytes): '1073741312'(Soft limit) and '-1'(Hard limit).
 GLPCTL121I Maximum Data Segment(Kbytes) soft ulimit for the process was 131072 and it is modified to the prescribed minimum 262144.
 GLPCTL119I Maximum File Size(512 bytes block) soft ulimit for the process is -1 and the prescribed minimum is 2097151.
 GLPCTL122I Maximum Open Files soft ulimit for the process is 2000 and the prescribed minimum is 500.
 GLPCTL121I Maximum Physical Memory(Kbytes) soft ulimit for the process was 32768 and it is modified to the prescribed minimum 262144.
 GLPCTL121I Maximum Stack Size(Kbytes) soft ulimit for the process was 32768 and it is modified to the prescribed minimum 65536.
 GLPCTL119I Maximum Virtual Memory(Kbytes) soft ulimit for the process is -1 and the prescribed minimum is 1048576.
 GLPCOM024I The extended Operation plugin is successfully loaded from libevent.a.
 GLPCOM024I The extended Operation plugin is successfully loaded from libtranext.a.
 GLPCOM024I The extended Operation plugin is successfully loaded from libldaprepl.a.
 GLPSRV155I The DIGEST-MD5 SASL Bind mechanism is enabled in the configuration file.
 GLPCOM021I The preoperation plugin is successfully loaded from libDigest.a.
 GLPCOM024I The extended Operation plugin is successfully loaded from libevent.a.
 GLPCOM024I The extended Operation plugin is successfully loaded from libtranext.a.
 GLPCOM023I The postoperation plugin is successfully loaded from libpsearch.a.
 GLPCOM024I The extended Operation plugin is successfully loaded from libpsearch.a.
 GLPCOM025I The audit plugin is successfully loaded from libldapaudit.a.
 GLPCOM025I The audit plugin is successfully loaded from /usr/ccs/lib/libsecldapaudit64.a(shr.o).
 GLPCOM024I The extended Operation plugin is successfully loaded from libevent.a.
 GLPCOM023I The postoperation plugin is successfully loaded from libpsearch.a.
 GLPCOM024I The extended Operation plugin is successfully loaded from libpsearch.a.
 GLPCOM022I The database plugin is successfully loaded from libback-config.a.
 GLPCOM024I The extended Operation plugin is successfully loaded from libevent.a.
 GLPCOM024I The extended Operation plugin is successfully loaded from libtranext.a.
 GLPCOM023I The postoperation plugin is successfully loaded from libpsearch.a.
 GLPCOM024I The extended Operation plugin is successfully loaded from libpsearch.a.
 GLPCOM022I The database plugin is successfully loaded from libback-rdbm.a.
 GLPCOM010I Replication plugin is successfully loaded from libldaprepl.a.
 GLPCOM021I The preoperation plugin is successfully loaded from libpta.a.
 GLPSRV017I Server configured for secure connections only.
 GLPSRV015I Server configured to use 636 as the secure port.
 GLPCOM024I The extended Operation plugin is successfully loaded from libloga.a.

GLPCOM024I The extended Operation plugin is successfully loaded from libidsfget.a.
GLPSRV180I Pass-through authentication is disabled.
GLPCOM004I SSL port initialized to 636.
Migrating users and groups to LDAP server.
#

At this point a query can be issued to the LDAP server to test its functionality. The **ldapsearch** command is used to retrieve information from the LDAP server and to execute an SSL search on the server that was just started. It can be used in the following way:

```
/opt/IBM/ldap/V6.1/bin/ldapsearch -D cn=admin -w passwd -h localhost -Z -K
/etc/ldap/ldap_server.kdb -p 636 -b "cn=SSL,cn=Configuration" "(ibm-slapdSslAuth=*)"
cn=SSL, cn=Configuration
cn=SSL
ibm-slapdSecurePort=636
ibm-slapdSecurity=SSLOnly
ibm-slapdSslAuth=serverauth
ibm-slapdSslCertificate=none
ibm-slapdSslCipherSpec=AES
ibm-slapdSslCipherSpec=AES-128
ibm-slapdSslCipherSpec=RC4-128-MD5
ibm-slapdSslCipherSpec=RC4-128-SHA
ibm-slapdSslCipherSpec=TripleDES-168
ibm-slapdSslCipherSpec=DES-56
ibm-slapdSslCipherSpec=RC4-40-MD5
ibm-slapdSslCipherSpec=RC2-40-MD5
ibm-slapdSslFIPSProcessingMode=false
ibm-slapdSslKeyDatabase=/etc/ldap/ldap_server.kdb
ibm-slapdSslKeyDatabasePW={AES256}31Ip2qH5pLx0IPX9NTbgvA==
ibm-slapdSslPKCS11AcceleratorMode=none
ibm-slapdSslPKCS11Enabled=false
ibm-slapdSslPKCS11Keystorage=false
ibm-slapdSslPKCS11Lib=libcknfast.so
ibm-slapdSslPKCS11TokenLabel=none
objectclass=top
objectclass=ibm-slapdConfigEntry
objectclass=ibm-slapdSSL
```

In this example, the SSL configuration is retrieved from the server. Note that the database key password is stored in a cryptographic form:

({AES256}31Ip2qH5pLx0IPX9NTbgvA==) .

After the LDAP server has been shown to be working, the Virtual I/O Server can be configured as a client.

Configuring the Virtual I/O Server as an LDAP client

The first thing to be checked on the Virtual I/O Server before configuring it as a secure LDAP client is whether the `ldap.max_crypto_client` file sets are installed. To check this, issue the `ls1pp` command on the Virtual I/O Server as root as follows:

```
# ls1pp -l |grep ldap
ldap.client.adt          5.2.0.0  COMMITTED  Directory Client SDK
ldap.client.rte          5.2.0.0  COMMITTED  Directory Client Runtime (No
ldap.max_crypto_client.adt
ldap.max_crypto_client.rte
ldap.client.rte          5.2.0.0  COMMITTED  Directory Client Runtime (No
```

If the file sets are not installed, proceed with the installation before going forward with these steps. These file sets can be found on the Virtual I/O Server Expansion Pack media. The Expansion Pack media comes with the Virtual I/O Server Version installation media.

Transfer the database key from the LDAP server to the Virtual I/O Server. In this example, `ldap_server.kdb` and `ldap_server.sth` were transferred from `/etc/ldap` on the LDAP server to `/etc/ldap` on the Virtual I/O Server.

On the Virtual I/O Server, the `mkldap` command is used to configure it as an LDAP client. To configure the Virtual I/O Server as a secure LDAP client of the LDAP server that was previously configured, use the following command:

```
$ mkldap -bind cn=admin -passwd passw0rd -host NIM_server -base o=ibm -keypath
/etc/ldap/ldap_server.kdb -keypasswd passw0rd -port 636
gskjs.rte
gskjt.rte
gksa.rte
gskta.rte
```

To check whether the secure LDAP configuration is working, create an LDAP user using the `mkuser` command with the `-ldap` flag, and then use the `lsuser` command to check its characteristics as shown in Example 16-12. Note that the registry of the user is now stored on the LDAP server.

Example 16-12 Creating an ldap user on the Virtual I/O Server

```
$ mkuser -ldap itso
itso's Old password:
itso's New password:
Enter the new password again:
$ lsuser itso
itso roles=Admin account_locked=false expires=0 histexpire=0 histsize=0
loginretries=0 maxage=0 maxexpired=-1 maxrepeats=8 minage=0 minalpha=0
mindiff=0 minlen=0 minother=0 pwdwarntime=330 registry=LDAP SYSTEM=LDAP
```

When the user `itso` tries to log in, its password has to be changed as shown in Example 16-13.

Example 16-13 Log on to the Virtual I/O Server using an LDAP user

```
login as: itso
itso@9.3.5.108's password:
[LDAP]: 3004-610 You are required to change your password.
        Please choose a new one.
WARNING: Your password has expired.
You must change your password now and login again!
Changing password for "itso"
itso's Old password:
itso's New password:
Enter the new password again:
```

Another way to test whether the configuration is working is to use the `ldapsearch` command to do a search on the LDAP directory. In Example 16-14, this command is used to search for the characteristics of the `o=ibm` object.

Example 16-14 Searching the LDAP server

```
$ ldapsearch -b o=ibm -h NIM_server -D cn=admin -w passw0rd -s base -p 636 -K
/etc/ldap/ldap_server.kdb -N ldap_server -P passw0rd objectclass=*
o=ibm
objectclass=top
objectclass=organization
o=ibm
```

The secure LDAP connection between the LDAP server and the Virtual I/O Server is now configured and operational.

Network Time Protocol configuration

A synchronized time is important for error logging, Kerberos, and various monitoring tools. The Virtual I/O Server has an NTP client installed. To configure it you can create or edit the configuration file `/home/padmin/config/ntp.conf` using the following command as shown in Example 16-15:

```
$ vi /home/padmin/config/ntp.conf
```

Example 16-15 Content of the `/home/padmin/config/ntp.conf` file

```
server ptbtime1.ptb.de
server ptbtime2.ptb.de
driftfile /home/padmin/config/ntp.drift
tracefile /home/padmin/config/ntp.trace
logfile /home/padmin/config/ntp.log
```

After it is configured, you start the `xntpd` service using the **startnetsvc** command as shown in Example 16-16.

Example 16-16 Start of the xntpd daemon

```
$ startnetsvc xntpd
0513-059 The xntpd Subsystem has been started. Subsystem PID is 123092.
```

After the daemon is started, check your `ntp.log` file. If it shows messages similar to those in Example 16-17, you have to set the time manually first.

Example 16-17 Too large time error

```
$ cat config/ntp.log
5 Dec 13:52:26 xntpd[516180]: SRC stop issued.
5 Dec 13:52:26 xntpd[516180]: exiting.
5 Dec 13:56:57 xntpd[516188]: synchronized to 9.3.4.7, stratum=3
5 Dec 13:56:57 xntpd[516188]: time error 3637.530348 is way too large (set
clock manually)
```

In order to set the date on the Virtual I/O Server, use the **chdate** command:

```
$ chdate 1206093607
$ Thu Dec 6 09:36:16 CST 2007
```

If the synchronization is successful, your log in `/home/padmin/config/ntp.log` should look like Example 16-18.

Example 16-18 Successful ntp synchronization

```
6 Dec 09:48:55 xntpd[581870]: synchronized to 9.3.4.7, stratum=2
6 Dec 10:05:34 xntpd[581870]: time reset (step) 998.397993 s
6 Dec 10:05:34 xntpd[581870]: synchronisation lost
6 Dec 10:10:54 xntpd[581870]: synchronized to 9.3.4.7, stratum=2
```

Remember: In Virtual I/O Server version 1.5.2.0 and earlier, the default configuration file used by the **startnetsvc xntpd** command, and the `rc.tcpip` startup file can differ. This might cause unpredictable results when rebooting a partition. See APAR IZ13781. In subsequent releases the default file does not differ.

Setting up Kerberos on the Virtual I/O Server

In order to use Kerberos on the Virtual I/O Server, you first have to install the Kerberos `krb5.client.rte` file set from the Virtual I/O Server Expansion Pack.

You then have to insert the first expansion pack media in the DVD drive. In case the drive is mapped for the other partitions to access it, you have to unmap it on the Virtual I/O Server with the **rmvdev** command, as follows:

```
$ lsmap -all | grep cd
Backing device          cd0
$ rmvdev -vdev cd0
vtopt0 deleted
```

You can then run the **installp** command. We use the **oem_setup_env** command to do this because **installp** must run with the root login.

```
$ echo "installp -agXYd /dev/cd0 krb5.client.rte" | oem_setup_env
+-----+
                        Pre-deinstall Verification...
+-----+
Verifying selections...done
```

[output part removed for clarity purpose]

Installation Summary

Name	Level	Part	Event	Result
krb5.client.rte	1.4.0.3	USR	APPLY	SUCCESS
krb5.client.rte	1.4.0.3	ROOT	APPLY	SUCCESS

The Kerberos client file sets are now installed on the Virtual I/O Server. The login process to the operating system remains unchanged. Therefore, you must configure the system to use Kerberos as the primary means of user authentication.

To configure the Virtual I/O Server to use Kerberos as the primary means of user authentication, run the **mkkrb5clnt** command with the following parameters:

```
$ oem_setup_env
# mkkrb5clnt -c KDC -r realm -a admin -s server -d domain -A -i database -K -T
# exit
```

The **mkkrb5clnt** command parameters are:

- c Sets the Kerberos Key Center (KDC) that centralizes authorizations.
- r Sets the Kerberos realm.
- s Sets the Kerberos admin server.
- K Specifies Kerberos to be configured as the default authentication scheme.
- T Specifies the flag to acquire server admin TGT based admin ticket.

For integrated login, the **-i** flag requires the name of the database being used. For LDAP, use the load module name that specifies LDAP. For local files, use the keyword files.

For example, to configure the VIO_Server1 Virtual I/O Server to use the ITSC.AUSTIN.IBM.COM realm, the krb_master admin and KDC server, the itsc.austin.ibm.com domain, and the local database, type the following:

```
$ oem_setup_env
# mkkrb5clnt -c krb_master.itsc.austin.ibm.com -r ITSC.AUSTIN.IBM.COM \
-s krb_master.itsc.austin.ibm.com -d itsc.austin.ibm.com -A -i files -K -T
```

```
Password for admin/admin@ITSC.AUSTIN.IBM.COM:
Configuring fully integrated login
Authenticating as principal admin/admin with existing credentials.
WARNING: no policy specified for host/VIO_Server1@ITSC.AUSTIN.IBM.COM;
defaulting to no policy. Note that policy may be overridden by
ACL restrictions.
Principal "host/VIO_Server1@ITSC.AUSTIN.IBM.COM" created.
```

```
Administration credentials NOT DESTROYED.
Making root a Kerberos administrator
Authenticating as principal admin/admin with existing credentials.
WARNING: no policy specified for root/VIO_Server1@ITSC.AUSTIN.IBM.COM;
defaulting to no policy. Note that policy may be overridden by
ACL restrictions.
Enter password for principal "root/VIO_Server1@ITSC.AUSTIN.IBM.COM":
Re-enter password for principal "root/VIO_Server1@ITSC.AUSTIN.IBM.COM":
Principal "root/VIO_Server1@ITSC.AUSTIN.IBM.COM" created.
```

```
Administration credentials NOT DESTROYED.
Configuring Kerberos as the default authentication scheme
Cleaning administrator credentials and exiting.
# exit
```

This example results in the following actions:

1. Creates the /etc/krb5/krb5.conf file. Values for realm name, Kerberos admin server, and domain name are set as specified on the command line. Also, this updates the paths for the default_keytab_name, kdc, and kadmin log files.
2. The **-i** flag configures fully integrated login. The database entered is the location where AIX user identification information is stored. This is different than the Kerberos principal storage. The storage where Kerberos principals are stored is set during the Kerberos configuration.
3. The **-K** flag configures Kerberos as the default authentication scheme. This allows the users to become authenticated with Kerberos at login time.

4. The **-A** flag adds an entry in the Kerberos database to make root an admin user for Kerberos.
5. The **-T** flag acquires the server admin TGT-based admin ticket.

If a system is installed that is located in a separate DNS domain than the KDC, the following additional actions must be performed:

1. Edit the `/etc/krb5/krb5.conf` file and add another entry after `[domain realm]`.
2. Map the separate domain to your realm.

For example, if you want to include a client that is in the `abc.xyz.com` domain into your `MYREALM` realm, the `/etc/krb5/krb5.conf` file includes the following additional entry:

```
[domain realm]
    .abc.xyz.com = MYREALM
```

Managing users

When the Virtual I/O Server is installed, the only user type that is active is the prime administrator (`padmin`), which can create additional user IDs with the following roles:

- ▶ System administrator
- ▶ Service representative
- ▶ Development engineer

Restriction: You cannot create the prime administrator (`padmin`) user ID. It is automatically created and enabled after the Virtual I/O Server is installed.

Table 16-3 lists the user management tasks available on the Virtual I/O Server and the commands you must run to accomplish each task.

Table 16-3 Task and associated command to manage Virtual I/O Server users

Task	Command
Create a system administrator user ID	<code>mkuser</code>
Create a service representative (SR) user ID	<code>mkuser</code> with the <code>-sr</code> flag
Create a development engineer (DE) user ID	<code>mkuser</code> with the <code>-de</code> flag
Create a LDAP user	<code>mkuser</code> with the <code>-ldap</code> flag
List a user's attributes	<code>lsuser</code>
Change a user's attributes	<code>chuser</code>
Switch to another user	<code>su</code>
Remove a user	<code>rmuser</code>

Creating a system administrator account

In Example 16-19 we show how to create a system administration account with the default values and then check its attributes.

Example 16-19 Creating a system administrator user and checking its attributes

```
$ mkuser johng
johng's New password:
Enter the new password again:
$ lsuser johng
johng roles=Admin account_locked=false expires=0 histexpire=0 histsize=0
loginretries=0 maxage=0 maxexpired=-1 maxrepeats=8 minage=0 minalpha=0
mindiff=0 minlen=0 minother=0 pldwarntime=330 registry=files SYSTEM=compat
```

The system administrator account has access to all commands except:

- ▶ **cleargcl**
- ▶ **lsfailedlogin**
- ▶ **lsgcl**
- ▶ **mirrorios**
- ▶ **mkuser**
- ▶ **oem_setup_env**
- ▶ **rmuser**
- ▶ **shutdown**
- ▶ **unmirrorios**

Creating a service representative (SR) account

In Example 16-20, we have created a service representative (SR) account. This type of account enables a service representative to run commands required to service the system without being logged in as root. This includes the following command types:

- ▶ Run diagnostics, including service aids (for example, hot plug tasks, certify, format, and so forth).
- ▶ Run all commands that can be run by a group system.
- ▶ Configure and unconfigure devices that are not busy.
- ▶ Use the service aid to update the system microcode.
- ▶ Perform the shutdown and reboot operations.

The preferred SR login user name is qserv.

Example 16-20 Creating a service representative account

```
$ mkuser -sr qserv
qserv's New password:
Enter the new password again:
```

```
$ lsuser qserv
qserv roles=SRUser account_locked=false expires=0 histexpire=0 histsize=0
loginretries=0 maxage=0 maxexpired=-1 maxrepeats=8 minage=0 minalpha=0
mindiff=0 minlen=0 minother=0 pwdwarntime=330 registry=files SYSTEM=compat
```

When the service representative user logs in to the system for the first time, it is asked to change its password. After changing it, the diag menu is automatically loaded. It can then execute any task from that menu, or get out of it and execute commands on the command line.

Creating a read-only account

The Virtual I/O Server **mkuser** command allows read-only accounts to be created. Read-only accounts are able to view everything a system administrator account can view, but they cannot change anything. Auditors are usually given read-only accounts. Read-only accounts are created by padmin with the following command:

```
$ mkuser -attr prgr=view auditor
```

Tip: A read-only account will not be able to even write on its own home directory, but it can view all configuration settings.

Checking the global command log (gcl)

After the users and their roles are set up, it is important to periodically check what they have been doing on the Virtual I/O Server. We accomplish this with the **ls gcl** command.

The **ls gcl** command lists the contents of the global command log (gcl). This log contains a listing of all commands that have been executed by all Virtual I/O Server users. Each listing contains the date and time of execution, and the user ID the command was executed from. Example 16-21 shows the output of this command on our Virtual I/O Server.

Restriction: The **ls gcl** command can only be executed by the prime administrator (padmin) user.

Example 16-21 ls gcl command output

```
Nov 16 2007, 17:12:26 padmin ioslevel
Nov 16 2007, 17:25:55 padmin updateios -accept -dev /dev/cd0
...
Nov 20 2007, 15:26:34 padmin uname -a
Nov 20 2007, 15:29:26 qserv diagmenu
Nov 20 2007, 16:16:11 padmin lsfailedlogin
Nov 20 2007, 16:25:51 padmin ls gcl
```

```
Nov 20 2007, 16:28:52 padmin passwd johng
Nov 20 2007, 16:30:40 johng lsmapi -all
Nov 20 2007, 16:30:53 johng lsmapi -vadaptor vhost0
Nov 20 2007, 16:32:11 padmin lsgcl
```

Role-based access control

With Virtual I/O Server Version 2.2, and later, a system administrator can define roles based on job functions in an organization by using role-based access control (RBAC).

A system administrator can use role-based access control (RBAC) to define roles for users in the Virtual I/O Server. A role confers a set of permissions or authorizations to the assigned user. Thus, a user can only perform a specific set of system functions depending on the access rights that user is given. For example, if the system administrator creates the role **UserManagement** with authorization to access user management commands (Example 16-24 on page 465) and assigns this role to a user, that user can manage users on the system but has no further access rights.

The benefits of using role-based access control with the Virtual I/O Server are as follows:

- ▶ Splitting system management functions
- ▶ Providing better security by granting only necessary access rights to users
- ▶ Implementing and enforcing system management and access control consistently
- ▶ Managing and auditing system functions with ease

Role-based access control is based on the concepts of *authorizations*, *roles*, and *privileges*. An overview of these concepts is provided in the following sections, followed by an example of using role-based access control.

Authorizations

The Virtual I/O Server creates authorizations that closely emulate the authorizations of the AIX operating system. The authorizations emulate naming conventions and descriptions, but are only applicable to the Virtual I/O Server specific requirements. By default, the **padmin** user is granted all the authorizations on the Virtual I/O Server, and can run all the commands. The other types of users (created by using the **mkuser** command) retain their command execution permissions.

The **mkauth** command creates a new user-defined authorization in the authorization database. You can create authorization hierarchies by using a dot (.) in the **auth** parameter to create an authorization of the form *ParentAuth.SubParentAuth.SubSubParentAuth....* All parent elements in the **auth** parameter must exist in the authorization database before the authorization is created. The maximum number of parent elements that you can use to create an authorization is eight.

You can set authorization attributes when you create authorizations through the **Attribute=Value** parameter. Every authorization that you create must have a value for the **id** authorization attribute. If you do not specify the **id** attribute using the **mkauth** command, the command automatically generates a unique ID for the authorization. If you specify an ID, the value must be unique and greater than 15000. The IDs 1 - 15000 are reserved for system-defined authorizations.

The system-defined authorizations in the Virtual I/O Server start with **vios.** Hence, user-defined authorizations must not start with **vios.** or **aix.** Because the authorizations that start with **vios.** and **aix.** are considered system-defined authorizations, users cannot add any further hierarchies to these authorizations.

Unlike in the AIX operating system, users cannot create authorizations for all Virtual I/O Server commands. In the AIX operating system, an authorized user can create a hierarchy of authorizations for all the commands. However, in the Virtual I/O Server, authorizations can only be created for the commands or scripts owned by the user. Users cannot create any authorizations that start with **vios.** or **aix.** because they are considered system-defined authorizations. Hence, users cannot add any further hierarchies to these authorizations.

Authorization names must not begin with a dash (-), plus sign (+), at sign (@), or tilde (~). They must not contain spaces, tabs, or newline characters. You cannot use the keywords **ALL**, **default**, **ALLOW_OWNER**, **ALLOW_GROUP**, **ALLOW_ALL**, or an asterisk (*) as an authorization name. Do not use the following characters within an authorization string:

- ▶ : (colon)
- ▶ " (quotation mark)
- ▶ # (number sign)
- ▶ , (comma)
- ▶ = (equal sign)
- ▶ \ (backslash)
- ▶ / (forward slash)
- ▶ ? (question mark)
- ▶ ' (single quotation mark)
- ▶ ` (grave accent)

Table 16-4 lists the authorizations corresponding to the Virtual I/O Server commands. The `vios` and subsequent child authorizations, for example `vios` and `vios.device`, are not used. If a user is given a role that has either the parent or subsequent child authorization, for example `vios` or `vios.device`, that user will have access to all the subsequent children authorizations and their related commands. For example, a role with the authorization `vios.device` gives the user access to all `vios.device.config` and `vios.device.manage` authorizations and their related commands.

Table 16-4 Authorizations corresponding to Virtual I/O Server commands

Command	Authorization
<code>activatevg</code>	<code>vios.lvm.manage.varyon</code>
<code>alert</code>	<code>vios.system.cluster.alert</code>
<code>alt_root_vg</code>	<code>vios.lvm.change.altrootvg</code>
<code>artexdiff</code>	<code>vios.system.rtxpert.diff</code>
<code>artexget</code>	<code>vios.system.rtxpert.get</code>
<code>artexlist</code>	<code>vios.system.rtxpert.list</code>
<code>artexmerge</code>	<code>vios.system.rtxpert.merge</code>
<code>artexset</code>	<code>vios.system.rtxpert.set</code>
<code>backup</code>	<code>vios.fs.backup</code>
<code>backupios</code>	<code>vios.install.backup</code>
<code>bootlist</code>	<code>vios.install.bootlist</code>
<code>cattracerpt</code>	<code>vios.system.trace.format</code>
<code>cfgassist</code>	<code>vios.security.cfgassist</code>
<code>cfgdev</code>	<code>vios.device.config</code>
<code>cfglnagg</code>	<code>vios.network.config.lnagg</code>
<code>cfgnamesrv</code>	<code>vios.system.dns</code>
<code>cfgsvc</code>	<code>vios.system.config.agent</code>
<code>chauth</code>	<code>vios.security.auth.change</code>
<code>chbdsp</code>	<code>vios.device.manage.backing.change</code>
<code>chdate</code>	<code>vios.system.config.date.change</code>
<code>chdev</code>	<code>vios.device.manage.change</code>

Command	Authorization
chkdev	vios.device.manage.check
chlang	vios.system.config.locale
chlv	vios.lvm.manage.change
chpath	vios.device.manage.path.change
chrep	vios.device.manage.repos.change
chrole	vios.security.role.change
chsp -default ^a	vios.device.manage.spool.change
chtcip	vios.network.tcpip.change
chuser	vios.security.user.change
chvg	vios.lvm.manage.change
chvopt	vios.device.manage.optical.change
cl_snmp	vios.security.manage.snmp.query
cleandisk	vios.system.cluster
cluster	vios.system.cluster.create
cplv	vios.lvm.manage.copy
cpvdi	vios.lvm.manage.copy
deactivatevg	vios.lvm.manage.varyoff
diagmenu	vios.system.diagnostics
dsmc	vios.system.manage.tsm
entstat	vios.network.stat.ent
errlog	vios.system.log.view
exportvg	vios.lvm.manage.export
extendlv	vios.lvm.manage.extend
extendvg	vios.lvm.manage.extend
fcstat	vios.network.stat.fc
fsck	vios.fs.check
hostmap	vios.system.config.address

Command	Authorization
hostname	vios.system.config.hostname
importvg	vios.lvm.manage.import
invscout	vios.system.firmware.scout
ioslevel	vios.system.level
ldapadd	vios.security.manage.ldap.add
ldapsearch	vios.security.manage.ldap.search
ldfware	vios.system.firmware.load
license	vios.system.license.view
license -accept	vios.system.license
loadopt	vios.device.manage.optical.load
loginmsg	vios.security.user.login.msg
lsauth	vios.security.auth.list
lsdev	vios.device.manage.list
lsfailedlogin	vios.security.user.login.fail
lsfware	vios.system.firmware.list
lsgcl	vios.security.log.list
lslparinfo	vios.system.lpar.list
lslv	vios.lvm.manage.list
lsmap	vios.device.manage.map.phyvirt
lsnetsvc	vios.network.service.list
lsnports	vios.device.manage.list
lspath	vios.device.manage.list
lspv	vios.device.manage.list
lsrep	vios.device.manage.repos.list
lsrole	vios.security.role.list
lssecattr	vios.security.cmd.list
lssp	vios.device.manage.spool.list

Command	Authorization
lssvc	vios.system.config.list
lssw	vios.system.software.list
lstcpip	vios.network.tcpip.list
lsuser ^b	vios.security.user.list
lsvg	vios.lvm.manage.list
lsvopt	vios.device.manage.optical.list
migratepv	vios.device.manage.migrate
mirrorios	vios.lvm.manage.mirrorios.create
mkauth	vios.security.auth.create
mkbdsp	vios.device.manage.backing.create
mkkrb5clnt	vios.security.manage.kerberos.create
mkldap	vios.security.manage.ldap.create
mklv	vios.lvm.manage.create
mklvcopy	vios.lvm.manage.mirror.create
mkpath	vios.device.manage.path.create
mkrep	vios.device.manage.repos.create
mkrole	vios.security.role.create
mksp	vios.device.manage.spool.create
mktcpip	vios.network.tcpip.config
mkuser	vios.security.user.create
mkvdev	vios.device.manage.create
mkvdev -lnagg	vios.device.manage.create.lnagg
mkvdev -sea	vios.device.manage.create.sea
mkvdev -vdev	vios.device.manage.create.virtualdisk
mkvdev -vlan	vios.device.manage.create.vlan
mkvg	vios.lvm.manage.create
mkvopt	vios.device.manage.optical.create

Command	Authorization
motd	vios.security.user.msg
mount	vios.fs.mount
netstat	vios.network.tcpip.list
optimizenet	vios.network.config.tune
oem_platform_level	vios.system.level
oem_setup_env	vios.oemsetupenv
passwd ^c	vios.security.passwd
pdump	vios.system.dump.platform
ping	vios.network.ping
postprocesssvc	vios.system.config.agent
prepdev	vios.device.config.prepare
redefvg	vios.lvm.manage.reorg
reducevg	vios.lvm.manage.change
refreshvlan	vios.network.config.refvlan
remote_management	vios.system.manage.remote
replphyvol	vios.device.manage.replace
restore	vios.fs.backup
restorevgstruct	vios.lvm.manage.restore
rmauth	vios.security.auth.remove
rmbdsp	vios.device.manage.backing.remove
rmdev	vios.device.manage.remove
rmlv	vios.lvm.manage.remove
rmlvcopy	vios.lvm.manage.mirror.remove
rmpath	vios.device.manage.path.remove
rmrep	vios.device.manage.repos.remove
rmrole	vios.security.role.remove
rmsecattr	vios.security.cmd.remove

Command	Authorization
rmosp	vios.device.manage.spool.remove
rmtcpip	vios.network.tcpip.remove
rmuser	vios.security.user.remove
rmvdev	vios.device.manage.remove
rmvopt	vios.device.manage.optical.remove
rolelist	vios.security.role.list
savevgstruct	vios.lvm.manage.save
save_base	vios.device.manage.saveinfo
seastat	vios.network.stat.sea
setkst	vios.security.kst.set
setsecattr	vios.security.cmd.set
showmount	vios.fs.mount.show
shutdown	vios.system.boot.shutdown
snap	vios.system.trace.format
snmp_info	vios.security.manage.snmp.info
snmpv3_ssw	vios.security.manage.snmp.switch
snmp_trap	vios.security.manage.snmp.trap
startnetsvc	vios.network.service.start
startsvc	vios.system.config.agent.start
startsysdump	vios.system.dump
stopnetsvc	vios.network.service.stop
stopsvc	vios.system.config.agent.stop
stoptrace	vios.system.trace.stop
svmon	vios.system.stat.memory
syncvg	vios.lvm.manage.sync
sysstat	vios.system.stat.list
topas	vios.system.config.topas

Command	Authorization
topasrec	vios.system.config.topasrec
tracepriv	vios.security.priv.trace
traceroute	vios.network.route.trace
uname	vios.system.uname
unloadopt	vios.device.manage.optical.unload
unmirrorios	vios.lvm.manage.mirrorios.remove
unmount	vios.fs.unmount
updateios	vios.install
vasistat	vios.network.stat.vasi
vfcmap	vios.device.manage.map.virt
viosbr	vios.system.backup.cfg
viosbr -view	vios.system.backup.cfg.view
viosecure	vios.security.manage.firewall
viostat	vios.system.stat.io
vmstat	vios.system.stat.memory
wkldagent	vios.system.manage.workload.agent
wkldmgr	vios.system.manage.workload.manager
wkldout	vios.system.manage.workload.process

- a. Other options of the chsp command can be run by all.
- b. Any user can run this command to view a minimal set of user attributes. However, only users with this authorization can view all the user attributes.
- c. Users can change their own password without having this authorization. This authorization is required only if the user wants to change the password of other users.

Roles

The Virtual I/O Server retains its current roles and will have the appropriate authorizations assigned to the roles. Additional roles that closely emulate the roles in the AIX operating system can be created. The roles emulate naming conventions and descriptions, but are only applicable to the Virtual I/O Server specific requirements. Users cannot view, use, or modify any of the default roles in the Virtual I/O Server.

The following roles are the default roles in the AIX operating system. These roles are unavailable to the Virtual I/O Server users, and are not displayed.

- ▶ AccountAdmin
- ▶ BackupRestore
- ▶ DomainAdmin
- ▶ FSAdmin
- ▶ SecPolicy
- ▶ SysBoot
- ▶ SysConfig
- ▶ isso
- ▶ sa
- ▶ so

The following roles are the default roles in the Virtual I/O Server:

- ▶ Admin
- ▶ DEUser
- ▶ PAdmin
- ▶ RunDiagnostics
- ▶ SRUser
- ▶ SYSAdm
- ▶ ViewOnly

The **mkrole** command creates a role. The **newrole** parameter must be a unique role name. You cannot use the **ALL** or **default** keywords as the role name. Every role must have a unique role ID that is used for security decisions. If you do not specify the **id** attribute when you create a role, the **mkrole** command automatically assigns a unique ID to the role.

There is no standard naming convention for roles. However, existing names of roles cannot be used for creating roles.

The role parameter cannot contain spaces, tabs, or newline characters. To prevent inconsistencies, restrict role names to characters in the POSIX portable file name character set. You cannot use the keywords **ALL** or **default** as a role name. Do not use the following characters within a role-name string:

- ▶ : (colon)
- ▶ " (quotation mark)
- ▶ # (number sign)
- ▶ , (comma)
- ▶ = (equal sign)
- ▶ \ (backslash)
- ▶ / (forward slash)
- ▶ ? (question mark)
- ▶ ' (single quotation mark)
- ▶ ` (grave accent)

Privileges

A Privilege is an attribute of a process through which the process can bypass specific restrictions and limitations of the system. Privileges are associated with a process, and are acquired by running a privileged command. Privileges are defined as bit-masks in the operating system kernel and enforce access control over privileged operations. For example, the privilege bit PV_KER_TIME might control the kernel operation to modify the system date and time. Nearly 80 privileges are included with the operating system kernel and provide granular control over privileged operations. You can acquire the least privilege required to perform an operation through division of privileged operations in the kernel. This feature leads to enhanced security because a process hacker can only get access to one or two privileges in the system, and not to root user privileges.

Authorizations and roles are a user-level tool to configure user access to privileged operations. Privileges are the restriction mechanism used in the operating system kernel to determine if a process has authorization to perform an action. Hence, if a user is in a role session that has an authorization to run a command, and that command is run, a set of privileges are assigned to the process. There is no direct mapping of authorizations and roles to privileges. Access to several commands can be provided through an authorization. Each of those commands can be granted a different set of privileges.

Using role-based access control

Table 16-5 lists the commands related to role-based access control (RBAC).

Table 16-5 RBAC commands and their descriptions

Command	Description
chauth	Modifies attributes of the authorization that is identified by the <i>newauth</i> parameter
chrole	Changes attributes of the role identified by the <i>role</i> parameter
lsauth	Displays attributes of user-defined and system-defined authorizations from the authorization database
lsrole	Displays the role attributes
lssecattr	Lists the security attributes of one or more commands, devices, or processes
mkauth	Creates new user-defined authorizations in the authorization database
mkrole	Creates new roles
rmauth	Removes the user-defined authorization identified by the <i>auth</i> parameter

Command	Description
rmrole	Removes the role identified by the <i>role</i> parameter from the roles database
rmsecattr	Removes the security attributes for a command, a device, or a file entry that is identified by the <i>Name</i> parameter from the appropriate database
rolelist	Provides role and authorization information to the caller about the roles assigned to them
setkst	Reads the security databases and loads the information from the databases into the kernel security tables
setsecattr	Sets the security attributes of the command, device, or process that are specified by the <i>Name</i> parameter
swrole	Creates a role session with the roles that are specified by the <i>Role</i> parameter
tracepriv	Records the privileges that a command attempts to use when the command is run

We will use some of these commands to create a new role. We will create a role called **UserAccessManagement** with authorization to access user management commands and assigns this role to a user, so that user can manage users on the system but has no further access rights. This scenario may arise in a business where user access management is administered by a single team across all operating systems. This team would need user access management rights on all operating systems, but perhaps no authority to do anything else.

This new role will be given access to the following commands:

- ▶ passwd
- ▶ chuser
- ▶ mkuser
- ▶ lsuser
- ▶ lsfailedlogin
- ▶ loginmsg
- ▶ motd
- ▶ rmuser

First, we create the new role using the **mkrole** command as shown in Example 16-22.

Example 16-22 Using the mkrole command

```
$ mkrole
authorizations='vios.security.passwd,vios.security.user.change,vios.security.us
er.create,vios.security.user.list,vios.security.user.login.fail,vios.security.u
ser.login.msg,vios.security.user.msg,vios.security.user.remove'
UserAccessManagement
```

The values for the **authorizations** parameter were obtained from Table 16-4 on page 455. To confirm that the role has been created correctly, we can use the **lsrole** command to display the role's attributes as shown in Example 16-23.

Example 16-23 Using the lsrole command

```
$ lsrole UserAccessManagement
UserAccessManagement
authorizations=vios.security.passwd,vios.security.user.change,vios.security.use
r.create,vios.security.user.list,vios.security.user.login.fail,vios.security.us
er.login.msg,vios.security.user.msg,vios.security.user.remove roletlist= groups=
visibility=1 screens=* dfltmsg= msgcat= auth_mode=INVOKER id=21
```

Next we create a new user (uam1), linking the new user to the newly created role (UserAccessManagement) using the **mkuser** command as shown in Example 16-24.

Example 16-24 Creating a new user linked to a role

```
$ mkuser -attr roles=UserAccessManagement uam1
uam1's New password:
Enter the new password again:
```

If we wanted to add existing users to the new role, we can use the **chuser** command.

To verify that the user is now linked to the role, use the **lsuser** command as shown in Example 16-25.

Example 16-25 Displaying a user's role

```
$ lsuser uam1
uam1 roles=UserAccessManagement default_roles= account_locked=false expires=0
histexpire=0 histsize=0 loginretries=0 maxage=0 maxexpired=-1 maxrepeats=8
minage=0 minalpha=0 mindiff=0 minlen=0 minother=0 pwdwarntime=330
registry=files SYSTEM=compat
```

If the new user logs on and attempts to execute any command other than the ones specified through the UserAccessManagement role, they will receive the error shown in Example 16-26.

Example 16-26 Access to run command is not valid message

```
$ ioslevel  
Access to run command is not valid.
```

16.2 Storage virtualization setup

This section gives you details on storage provisioning from Virtual I/O Server using VSCSI, Virtual Fibre Channel, Virtual Optical, Virtual Tape, and Shared Storage Pools. This section also includes details such as how to configure availability for storage in a PowerVM environment.

16.2.1 Virtual SCSI

Provisioning the storage to clients using Virtual SCSI can be done from Virtual I/O Server command line or HMC. See the following sections for detailed steps required to do so. We use the basic scenario that we have used in Figure 12-1 on page 312.

Defining virtual disks

Virtual disks can either be whole physical disks, logical volumes, or files. The physical disks can either be Power Systems internal disks or SAN attached disks. Virtual disks can be defined using the Hardware Management Console or the Virtual I/O Server.

The Hardware Management Console provides a graphical user interface which makes the creation and mapping of virtual disks very easy without requiring a login on the Virtual I/O Server. However some functionalities such as specifying a name for the virtual target devices, displaying LUN IDs of SAN storage devices or the removal of virtual disks cannot be done through the HMC.

Virtual I/O Server command-line interface can be used for creating and managing virtual disks. While it is not as easy to use as the Hardware Management Console graphical user interface it provides access to the full range of available commands and options for managing virtual disks.

The following section shows how virtual disks can be defined using the Virtual I/O Server. Reading through it, you will get a clear understanding of the individual steps required to create and map a virtual disk.

Using logical volumes

Creating and mapping logical volumes was previously covered in 12.3, “Defining virtual disks for client partitions” on page 345.

Using physical disks

Instead of creating logical volumes on the Virtual I/O Server and mapping them to its client partitions, Power Systems internal physical disks or SAN storage LUNs can also be directly mapped to client partitions as whole disks without going through the Virtual I/O Server’s logical volume management.

Considerations: For performance reasons and configuration simplicity consider mapping whole LUNs to the Virtual I/O Server’s client partitions when using SAN storage rather than to split a LUN into separate logical volumes.

You can verify the available hdisks with the **lspv** command. When used with no options, the **lspv** command displays all available hdisk devices (Example 16-27). If the **lspv -free** command is used, only the hdisks which are free to be used as backing devices are displayed. We will use the newly defined hdisk6 to hdisk9.

Example 16-27 Listing hdisks

# lspv			
hdisk0	00c1f170d7a97dec	rootvg	active
hdisk1	00c1f170e170ae72	None	
hdisk2	00c1f170e170c9cd	None	
hdisk3	00c1f170e170dac6	None	
hdisk4	00c1f17093dc5a63	None	
hdisk5	00c1f170e170fbb2	None	
hdisk6	none	None	
hdisk7	none	None	
hdisk8	none	None	
hdisk9	none	None	

Important: Take special care to verify that an hdisk you are going to use is really not in use already by a client partition, especially because PVIDs written by an AIX client are not displayed by default on the Virtual I/O Server, and IBM i, or Linux, do not even use PVIDs as known by AIX.

It is useful to be able to correlate the SAN storage LUN IDs to hdisk numbers on the Virtual I/O Server. For SAN storage devices attached by the default MPIO multipath device driver, such as the IBM System Storage DS5000 series, the **mpio_get_config -Av** command is available for a listing of LUN names. In our example we used the **pcmpath query device** command from the SDDPCM multipath device driver for DS8000 storage as shown in Example 16-28.

These commands are part of the storage device drivers and you will have to use the **oem_setup_env** command to access them.

Example 16-28 Listing of LUN to hdisk mapping

```
$ oem_setup_env
# pcmpath query device

Total Dual Active and Active/Asymmetric Devices : 4

DEV#: 6  DEVICE NAME: hdisk6  TYPE: 2107900  ALGORITHM:  Load Balance
SERIAL: 75BALB11011
=====
Path#      Adapter/Path Name      State    Mode    Select    Errors
  0          fscsi0/path0      CLOSE   NORMAL        0         0
  1          fscsi1/path1      CLOSE   NORMAL        0         0

DEV#: 7  DEVICE NAME: hdisk7  TYPE: 2107900  ALGORITHM:  Load Balance
SERIAL: 75BALB11012
=====
Path#      Adapter/Path Name      State    Mode    Select    Errors
  0          fscsi0/path0      CLOSE   NORMAL        0         0
  1          fscsi1/path1      CLOSE   NORMAL        0         0

DEV#: 8  DEVICE NAME: hdisk8  TYPE: 2107900  ALGORITHM:  Load Balance
SERIAL: 75BALB11013
=====
Path#      Adapter/Path Name      State    Mode    Select    Errors
  0          fscsi0/path0      CLOSE   NORMAL        0         0
  1          fscsi1/path1      CLOSE   NORMAL        0         0

DEV#: 9  DEVICE NAME: hdisk9  TYPE: 2107900  ALGORITHM:  Load Balance
SERIAL: 75BALB11014
=====
Path#      Adapter/Path Name      State    Mode    Select    Errors
  0          fscsi0/path0      CLOSE   NORMAL        0         0
```

You can also find the LUN number for an hdisk with the **lsdev -dev hdiskn -vpd** command, where *n* is the hdisk number as shown in Example 16-29.

Example 16-29 Finding the LUN number

```
$ lsdev -dev hdisk6 -vpd
  hdisk6
U789D.001.DQDYKYW-P1-C1-T1-W500507630410412C-L4010401100000000  IBM MPIO FC
2107
```

```
Manufacturer.....IBM
Machine Type and Model.....2107900
Serial Number.....75BALB11011
EC Level.....278
Device Specific.(Z0).....10
Device Specific.(Z1).....0201
Device Specific.(Z2).....075
Device Specific.(Z3).....29205
Device Specific.(Z4).....08
Device Specific.(Z5).....00
```

PLATFORM SPECIFIC

```
Name: disk
Node: disk
Device Type: block
```

These are the steps to map whole disks in the same way as in the previous section using the same virtual SCSI server adapters:

1. You can use the **lsdev -vpd** command to list the virtual slot numbers corresponding to vhost numbers as shown in Example 16-30.

Example 16-30 Listing of slot number to vhost mapping

```
$ lsdev -vpd|grep vhost
vhost4 U9117.MMA.101F170-V1-C90 Virtual SCSI Server Adapter
vhost3 U9117.MMA.101F170-V1-C50 Virtual SCSI Server Adapter
vhost2 U9117.MMA.101F170-V1-C40 Virtual SCSI Server Adapter
vhost1 U9117.MMA.101F170-V1-C30 Virtual SCSI Server Adapter
vhost0 U9117.MMA.101F170-V1-C20 Virtual SCSI Server Adapter
```

2. Define the SCSI mappings to create the virtual target devices that associate to the logical volume you have defined in the previous step. Based on Example 16-30 on page 469, we have five virtual SCSI server vhost devices on the Virtual I/O Server. Four of these vhost devices are the ones we use for mapping our disks to. We also map the physical DVD drive to a virtual SCSI server adapter vhost4 to be accessible for the client partitions and call it vcd as shown in Example 16-31.

Example 16-31 Mapping SAN disks and the DVD drive, cd0

```
$ mkvdev -vdev hdisk6 -vadapter vhost0 -dev vnimsrv_rvg
vnimsrv_rvg Available
$ mkvdev -vdev hdisk7 -vadapter vhost1 -dev vdbsrv_rvg
vdbsrv_rvg Available
$ mkvdev -vdev hdisk8 -vadapter vhost2 -dev vIBMi_LS
vIBMi_LS Available
$ mkvdev -vdev hdisk9 -vadapter vhost3 -dev vlinux
vlinux Available
$ mkvdev -vdev cd0 -vadapter vhost4 -dev vcd
vcd Available
```

Considerations:

1. The same concept applies when creating disks that are to be used as data volumes instead of boot volumes.
2. You can map data disks through the same vhost adapters that are used for rootvg. VSCSI connections operate at memory speed and each adapter can handle a large number of target devices.

Defining virtual disks using the HMC

Another alternative for command line is HMC for creating virtual disks.

Use the following steps to build the logical volumes required to create the virtual disk for the client partition's rootvg based on our basic scenario using the Hardware Management Console:

1. Log in to the Hardware Management Console using the hscroot user ID.
When you are logged in, select **Configuration** → **Virtual Storage Management** as shown in Figure 16-10.

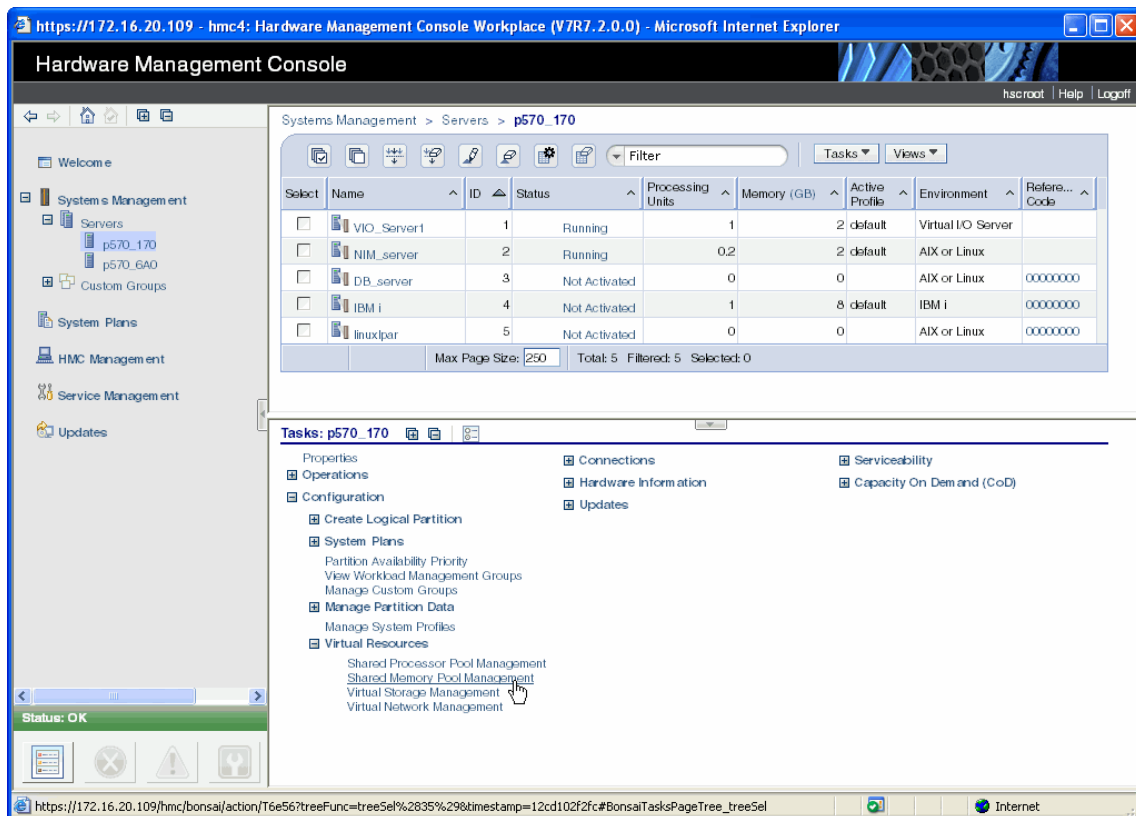


Figure 16-10 Starting the shared storage management HMC dialog

2. In the window that displays, click the **Query VIOS** button so that the Hardware Management Console queries the current storage configuration from the Virtual I/O Server.
3. Change to the **Storage Pools** tab and click the **Create storage pool** button as shown in Figure 16-11.

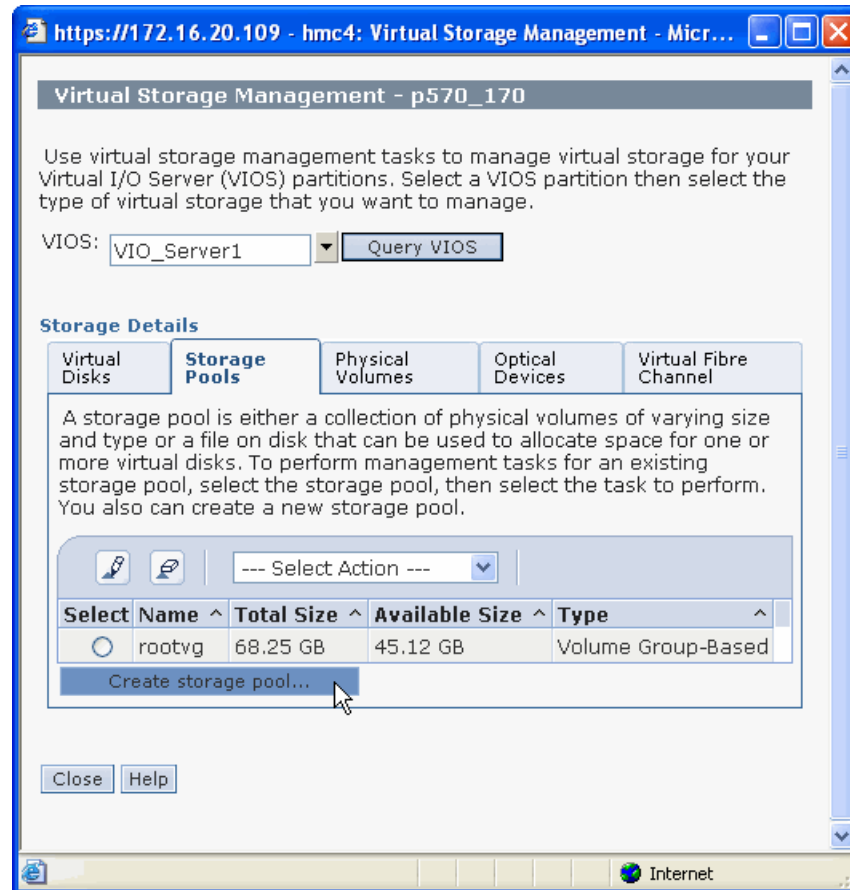


Figure 16-11 Creating a storage pool using the HMC

4. In the Create Storage Pool window, specify the **Storage pool name** and select the **Volume Group based** option from the **Storage pool type** pull-down menu. Then select the hdisks that have to be part of the storage pool.

Figure 16-12 shows the settings for our example configuration. After clicking **OK**, the storage pool will be created.

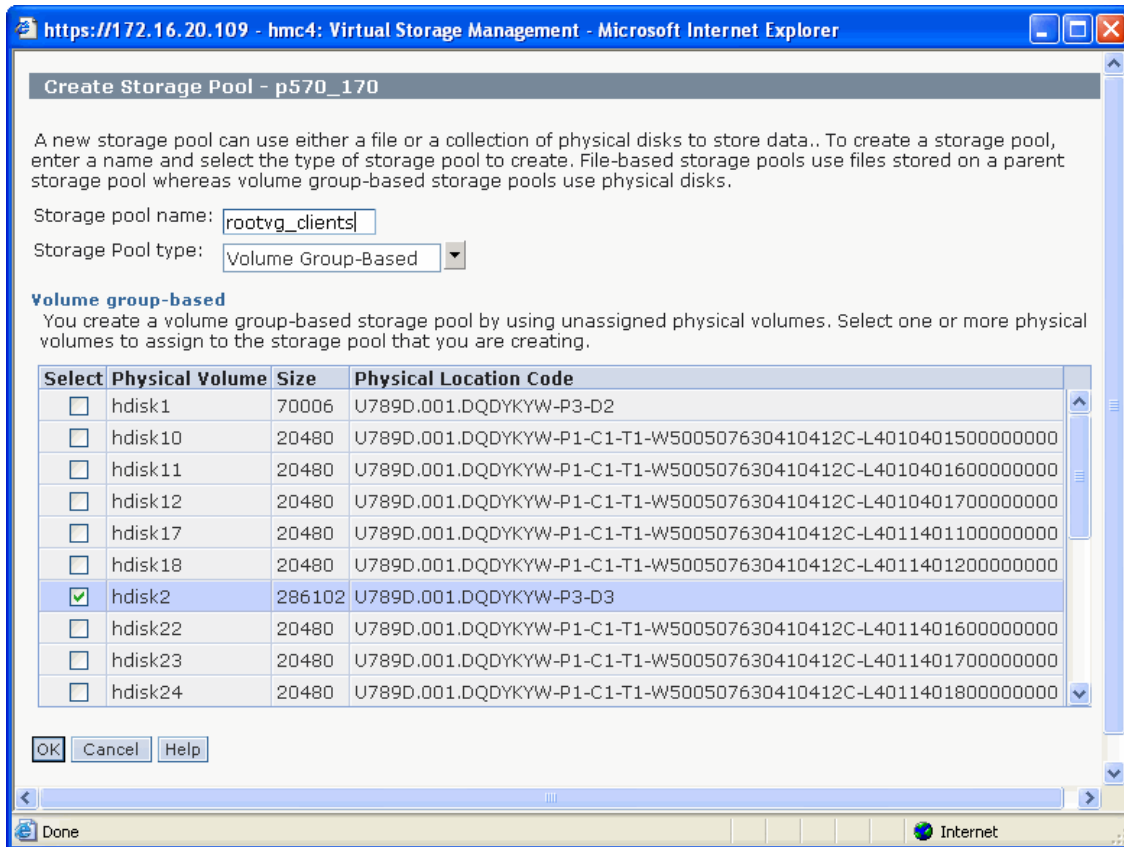


Figure 16-12 Defining storage pool attributes using the HMC GUI

5. Change to the **Virtual Disks** tab and start creating the virtual disks by clicking the **Create virtual disk** button. A window as shown in Figure 16-13 on page 474 will appear. You have to define the name of the virtual disk in the **Virtual disk name** field. Then select in which storage pool the virtual disk will be created by using the **Storage pool name** pull down menu. The size of the virtual disk is specified in the **Virtual disk size** field.

If the client partition that the virtual disks must be assigned to already exists, you can select it from the **Assigned partition** pull-down menu. If the partition does not yet exist, select **None** and assign the disk later.

Figure 16-13 shows the creation of a 10 GB virtual disk called `dbsrv_rvg` that will be assigned to partition `DB_server`.

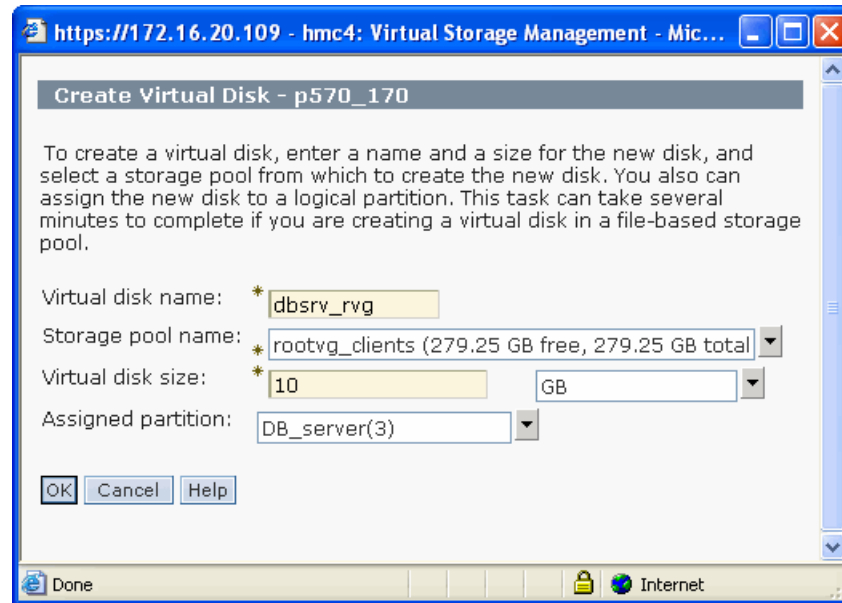


Figure 16-13 Creating a virtual disk using the HMC

16.2.2 Virtual Fibre Channel

This section describes how to configure SAN storage devices by Virtual Fibre Channel (VFC) for an AIX, IBM i or Linux client of the Virtual I/O Server. An IBM 2005-B32 SAN switch, an IBM Power Systems 570 server, and an IBM System Storage DS8300 storage system was used in our lab environment to describe the setup of the Virtual Fibre Channel environment.

The following configuration describes the setup of Virtual Fibre Channel in this section, as illustrated in Figure 16-14:

- ▶ A dedicated Virtual Fibre Channel server adapter (slot 31, 41, 51) is used in the Virtual I/O Server partition `VIO_Server1` for each virtual Fibre Channel client partition.
- ▶ Virtual Fibre Channel client adapter slots 31, 41 and 51 are used in the AIX, IBM i, and Linux virtual I/O client partitions.
- ▶ Each client partition accesses physical storage through its Virtual Fibre Channel adapter.

Figure 16-14 depicts the setup of Virtual Fibre Channel described in this section.

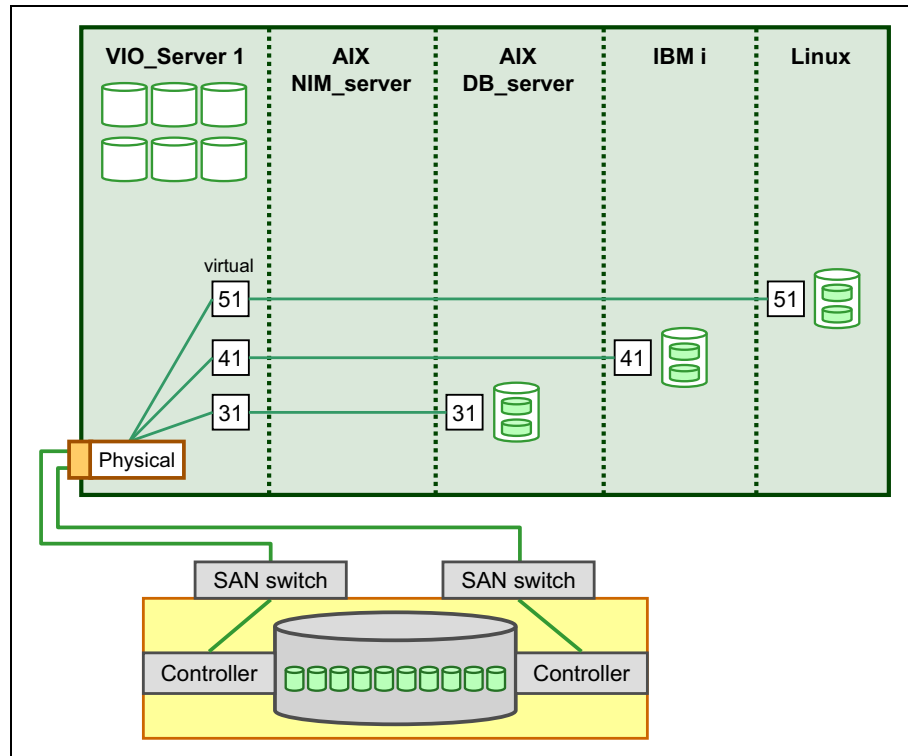


Figure 16-14 Virtual Fibre Channel adapter numbering

These steps describe how to set up the Virtual Fibre Channel environment:

1. On the SAN switch, you must perform two tasks before it can be used for Virtual Fibre Channel:
 - a. Update the firmware to a minimum level of Fabric OS (FOS) 5.3.0. To check the level of Fabric OS on the switch, log on to the switch and run the **version** command, as shown in Example 16-32:

Example 16-32 version command shows Fabric OS level

```
itso-aus-san-01:admin> version
Kernel:      2.6.14
Fabric OS:   v5.3.0
Made on:     Thu Jun 14 19:06:31 2007
Flash:       Tue Oct 13 12:30:07 2009
BootProm:    4.6.4
```

Reference: You can find the firmware for IBM SAN switches at:
<http://www-03.ibm.com/systems/storage/san/index.html>
 Click **Support** and select **Storage are network (SAN)** in the Product family. Then select your SAN product.

- b. After a successful firmware update, you must enable the Virtual Fibre Channel capability on each port of the SAN switch. Run the **portCfgNPiVPort** command to enable Virtual Fibre Channel, for example, for port 15 as follows:

```
itso-aus-san-01:admin> portCfgNPiVPort 15, 1
```

The **portcfgshow** command lists information for all ports, as shown in Example 16-33.

Example 16-33 List port configuration

```

itso-aus-san-01:admin> portcfgshow
Ports of Slot 0      0  1  2  3      4  5  6  7      8  9 10 11      12 13 14 15
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
Speed                AN AN AN AN      AN AN AN AN      AN AN AN AN      AN AN AN AN
Trunk Port           ON ON ON ON      ON ON ON ON      ON ON ON ON      ON ON ON ON
Long Distance        .. .. .. ..      .. .. .. ..      .. .. .. ..      .. .. .. ..
VC Link Init         .. .. .. ..      .. .. .. ..      .. .. .. ..      .. .. .. ..
Locked L_Port        .. .. .. ..      .. .. .. ..      .. .. .. ..      .. .. .. ..
Locked G_Port        .. .. .. ..      .. .. .. ..      .. .. .. ..      .. .. .. ..
Disabled E_Port      .. .. .. ..      .. .. .. ..      .. .. .. ..      .. .. .. ..
ISL R_RDY Mode       .. .. .. ..      .. .. .. ..      .. .. .. ..      .. .. .. ..
RSCN Suppressed      .. .. .. ..      .. .. .. ..      .. .. .. ..      .. .. .. ..
Persistent Disable... .. .. ..      .. .. .. ..      .. .. .. ..      .. .. .. ..
NPIV capability      ON ON ON ON      ON ON ON ON      ON ON ON ON      ON ON ON ON

      where AN:AutoNegotiate, ..:OFF, ?:INVALID,
      SN:Software controlled AutoNegotiation.

```

Tip: See your SAN switch users guide for the command to enable NPIV on your SAN switch.

2. Follow these steps to create the Virtual Fibre Channel server adapter in the Virtual I/O Server partition:
 - a. On the HMC, select the managed server to be configured:

Systems Management → Servers → <servername>
 - b. Select the Virtual I/O Server partition on which the Virtual Fibre Channel server adapter is to be configured. Then select from the tasks pop-up menu **Dynamic Logical Partitioning → Virtual Adapters** as shown in Figure 16-15.

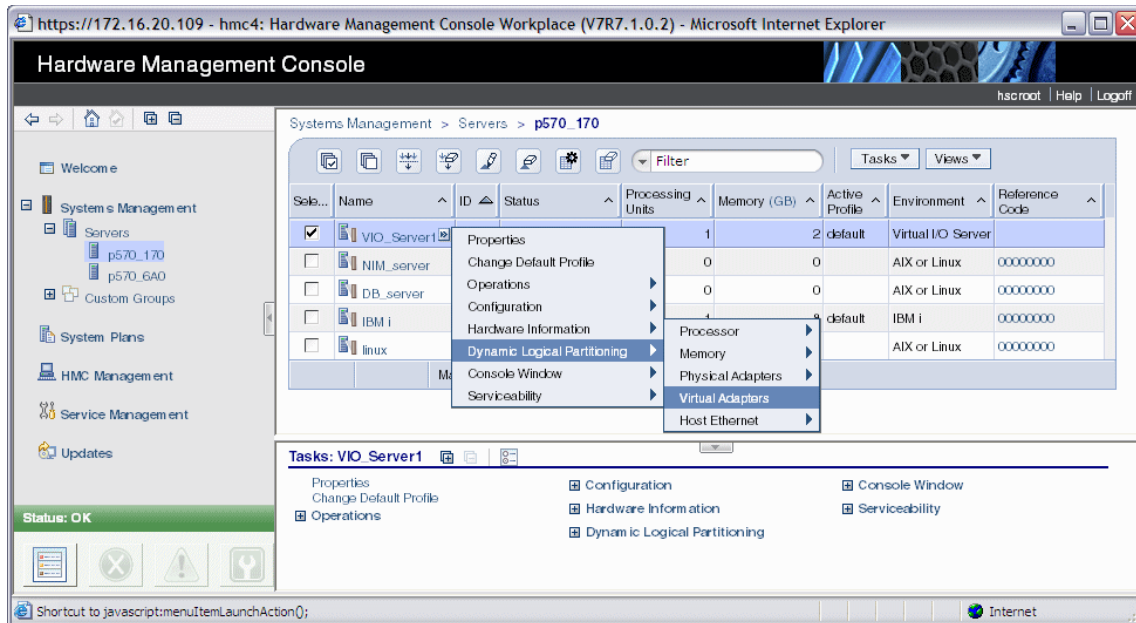


Figure 16-15 Dynamically add virtual adapter

- c. To create a virtual Fibre Channel server adapter, select **Actions** → **Create** → **Fibre Channel Adapter...** as shown in Figure 16-16.

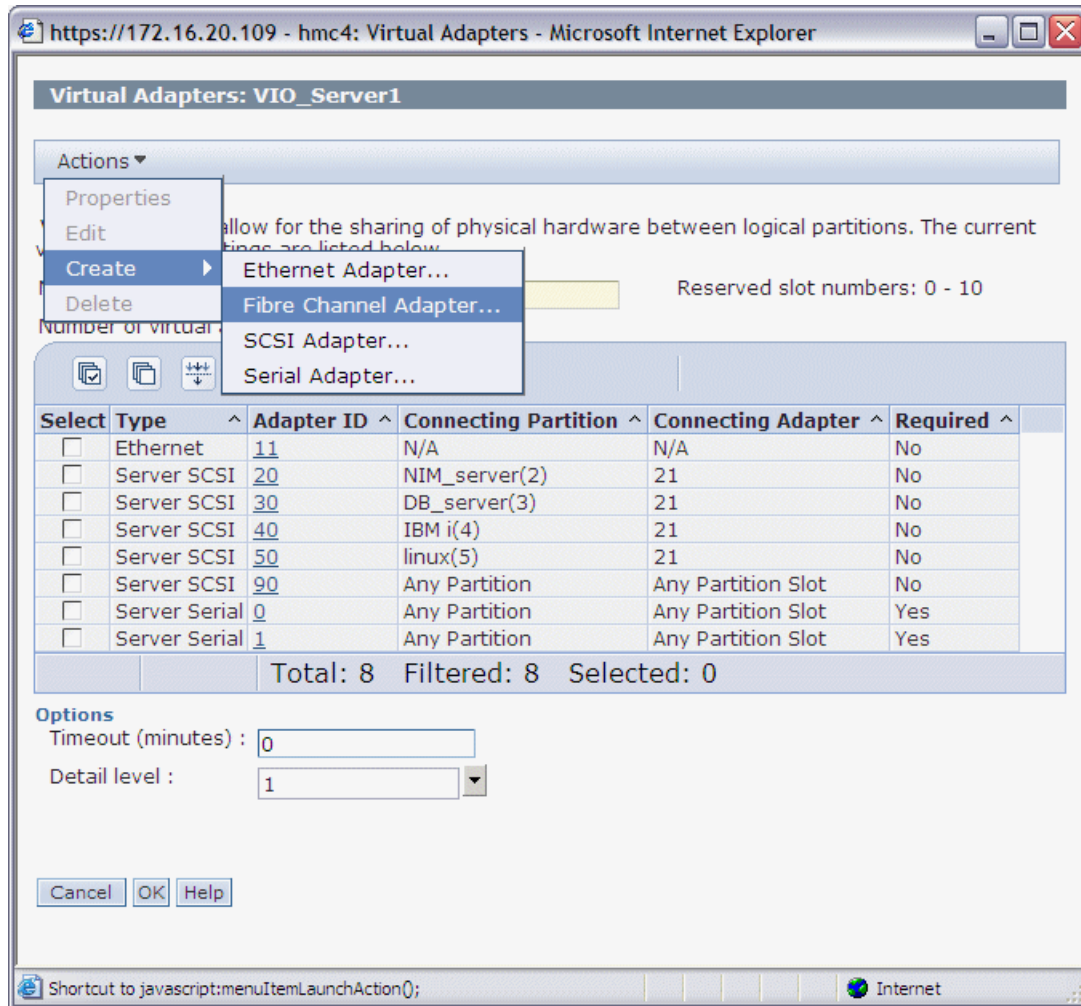


Figure 16-16 Create Fibre Channel server adapter

- d. Enter the virtual slot number for the Virtual Fibre Channel server adapter. Then select the Client Partition to which the adapter can be assigned, and enter the Client adapter ID as shown in Figure 16-17. Click **OK**.

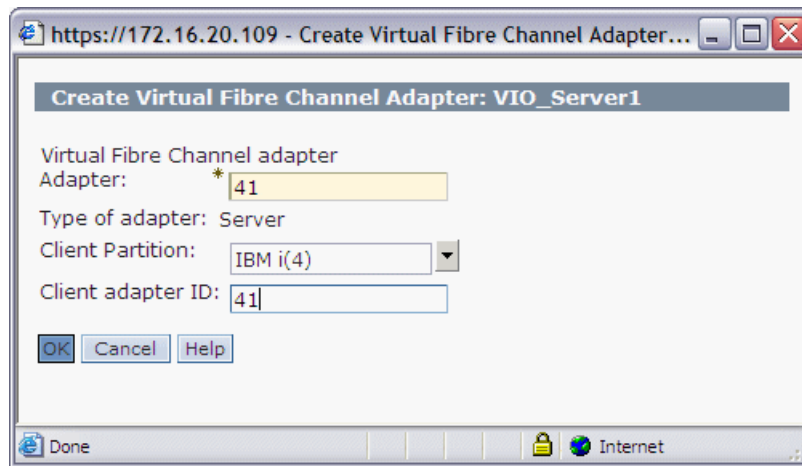


Figure 16-17 Set virtual adapter ID

- e. Click **OK** in the Virtual Adapters dialog to save the changes.

- f. Remember to update the partition profile of the Virtual I/O Server partition using the **Configuration** → **Save Current Configuration** option as shown in Figure 16-18 to save the changes to a new profile.

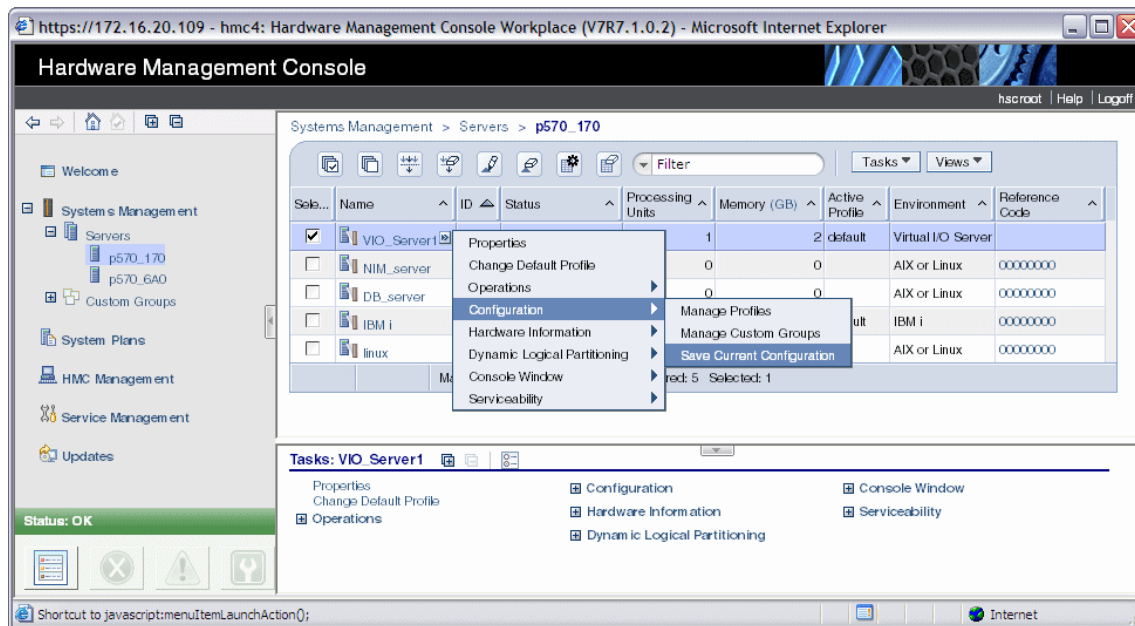


Figure 16-18 Save the Virtual I/O Server partition configuration

3. Follow these steps to create a Virtual Fibre Channel client adapter in the virtual I/O client partition.
 - a. Select the virtual I/O client partition on which the Virtual Fibre Channel client adapter is to be configured. Assuming the partition is not activated change the partition profile by selecting **Configuration** → **Manage Profiles** as shown in Figure 16-19.

Important: A Virtual Fibre Channel adapter can also be added to a running client partition using Dynamic Logical Partitioning (DLPAR), however notice that if then manually changing the partition profile to reflect the DLPAR change for trying to make it persistent across partition restarts, another *different* pair of virtual WWPNs will be generated. To prevent this undesired situation, which will require another SAN zoning and storage configuration change for the changed virtual WWPN to prevent an access loss condition, make sure to save any Virtual Fibre Channel client adapter DLPAR changes into a new partition profile by selecting **Configuration** → **Save Current Configuration** and change the default partition profile to the new profile.

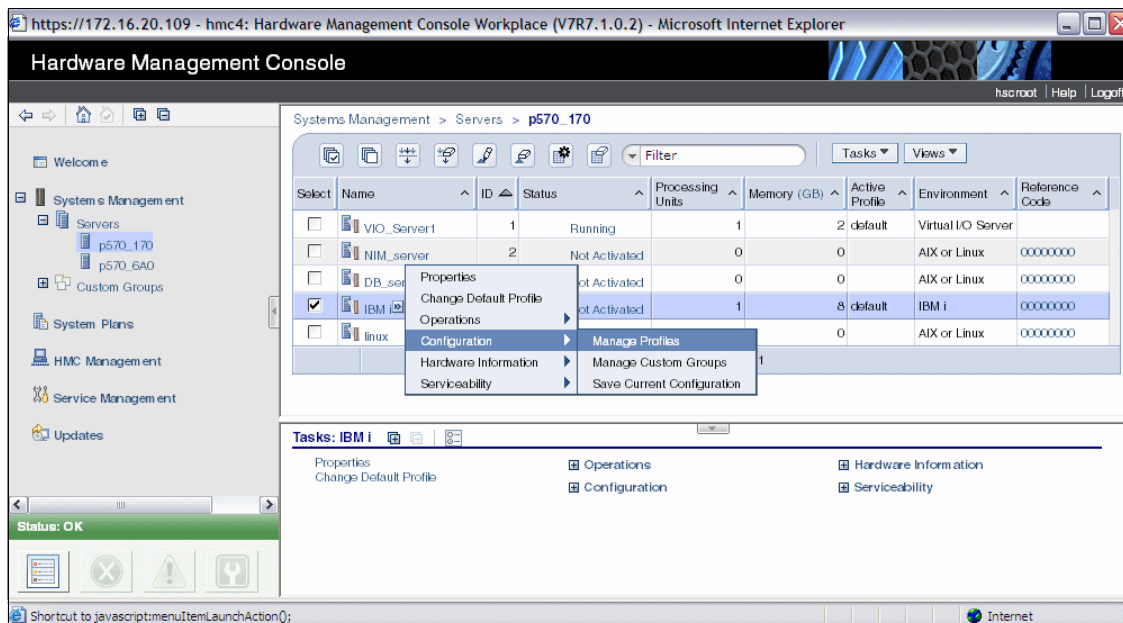


Figure 16-19 Change profile to add Virtual Fibre Channel client adapter

- b. Click the profile name to edit and select the **Virtual Adapters** tab in the Logical Partition Profile Properties dialog, then to create a virtual Fibre Channel client adapter, select **Actions** → **Create** → **Fibre Channel Adapter** as shown in Figure 16-20.

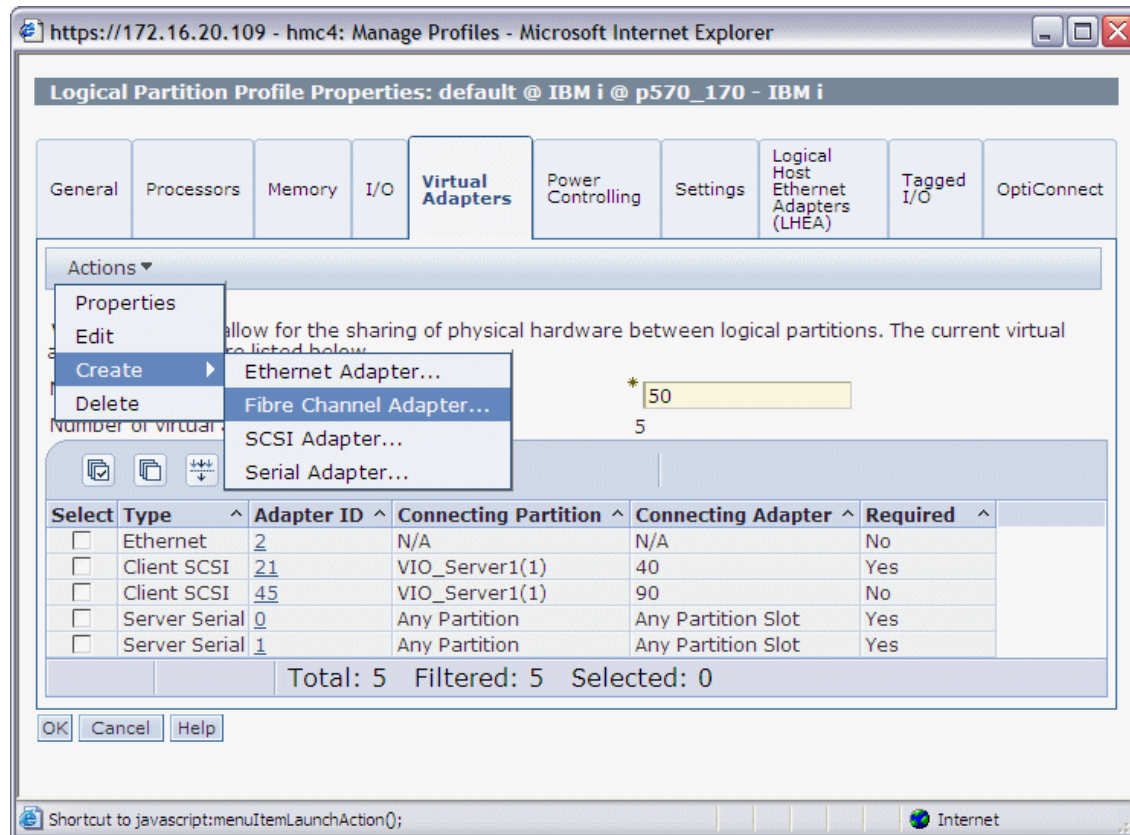


Figure 16-20 Create Fibre Channel client adapter

- c. Enter virtual slot number for the Virtual Fibre Channel client adapter. Then select the Virtual I/O Server partition to which the adapter can be assigned and enter the Server adapter ID as shown in Figure 16-21. Click **OK**.

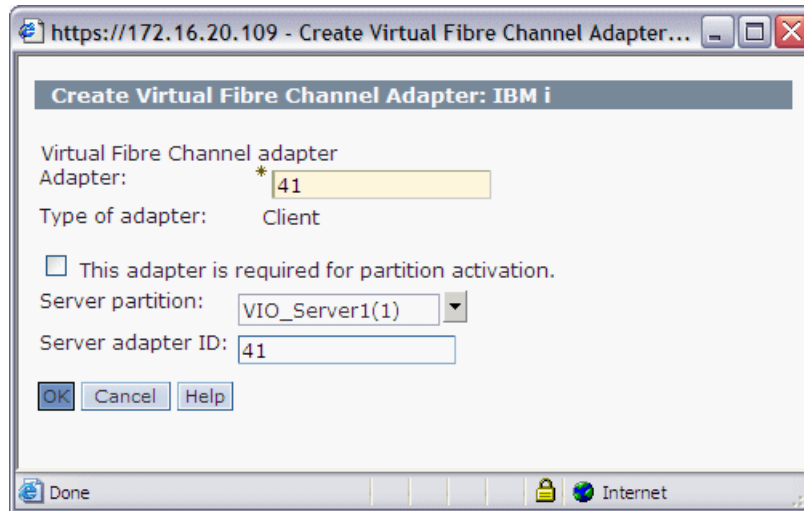


Figure 16-21 Define virtual adapter ID values

- d. Click **OK** and **Close** in the Managed Profiles dialog to save the changes.
4. Logon to the Virtual I/O Server partition as user padmin.
5. Run the **cfgdev** command to get the Virtual Fibre Channel server adapter(s) configured.
6. Run the command **lsdev -dev vfchost*** to list all available Virtual Fibre Channel server adapters in the Virtual I/O Server partition before mapping to a physical adapter, as shown in Example 16-34.

Example 16-34 lsdev -dev vfchost command on the Virtual I/O Server*

```
$ lsdev -dev vfchost*
name          status      description
vfchost0      Available   Virtual FC Server Adapter
```

7. The **lsdev -dev fcs*** command lists all available physical Fibre Channel server adapters in the Virtual I/O Server partition (Example 16-35).

Example 16-35 lsdev -dev fcs command on the Virtual I/O Server*

```
$ lsdev -dev fcs*
name          status      description
fcs0          Available   8Gb PCI Express Dual Port FC Adapter (df1000f114108a03)
fcs1          Available   8Gb PCI Express Dual Port FC Adapter (df1000f114108a03)
```

- Run the **lsnports** command to check the virtual Fibre Channel adapter readiness of the adapter and the SAN switch. Example 16-36 shows that the **fabric** attribute for the physical Fibre Channel adapter in slot C1 is set to 1. This means the adapter and the SAN switch are NPIV ready. If the value equals 0, then the adapter or SAN switch is not NPIV ready, and you need to check the SAN switch configuration.

Example 16-36 lsnports command on the Virtual I/O Server

\$ lsnports						
name	physloc	fabric	tports	aports	swwpns	awwpns
fcs0	U789D.001.DQDYKYW-P1-C1-T1	1	64	64	2048	2047
fcs1	U789D.001.DQDYKYW-P1-C1-T2	1	64	64	2048	2047

- Before mapping the virtual FC adapter to a physical adapter, get the **vfchost** name of the virtual adapter you created and the **fcs** name for the FC adapter from the previous **lsdev** commands output.
- To map the virtual Fibre Channel server adapter **vfchost0** to the physical Fibre Channel adapter **fcs0**, use the **vfcmmap** command as shown in Example 16-37.

Example 16-37 vfcmmap command with vfchost0 and fcs0

```
$ vfcmmap -vadapter vfchost0 -fcp fcs0
vfchost0 changed
```

- To list the mappings use the **lsmap -all -npiv** command, as shown in Example 16-38.

Example 16-38 lsmap -npiv -vadapter vfchost0 command

```
$ lsmap -all -npiv
Name          Physloc                      CIntID CIntName          CIntOS
-----
vfchost0      U9117.MMA.101F170-V1-C41      4 IBM i             IBM i

Status:LOGGED_IN
FC name:fcs0                      FC loc code:U789D.001.DQDYKYW-P1-C1-T1
Ports logged in:1
Flags:a<LOGGED_IN,STRIP_MERGE>
VFC client name:DC04              VFC client DRC:U9117.MMA.101F170-V4-C41
```

12. After you have created the virtual Fibre Channel server adapters in the Virtual I/O Server partition and in the virtual I/O client partition, you need to do the correct zoning in the SAN switch. Follow the next steps:
 - a. Get the information about the WWPN of the virtual Fibre Channel client adapter created in the virtual I/O client partition.
 - i. Select the appropriate virtual I/O client partition, then from the task popup-menu click **Properties**. Expand the **Virtual Adapters** tab, select the Client Fibre Channel client adapter and then select **Actions** → **Properties** to list the properties of the virtual Fibre Channel client adapter, as shown in Figure 16-22.

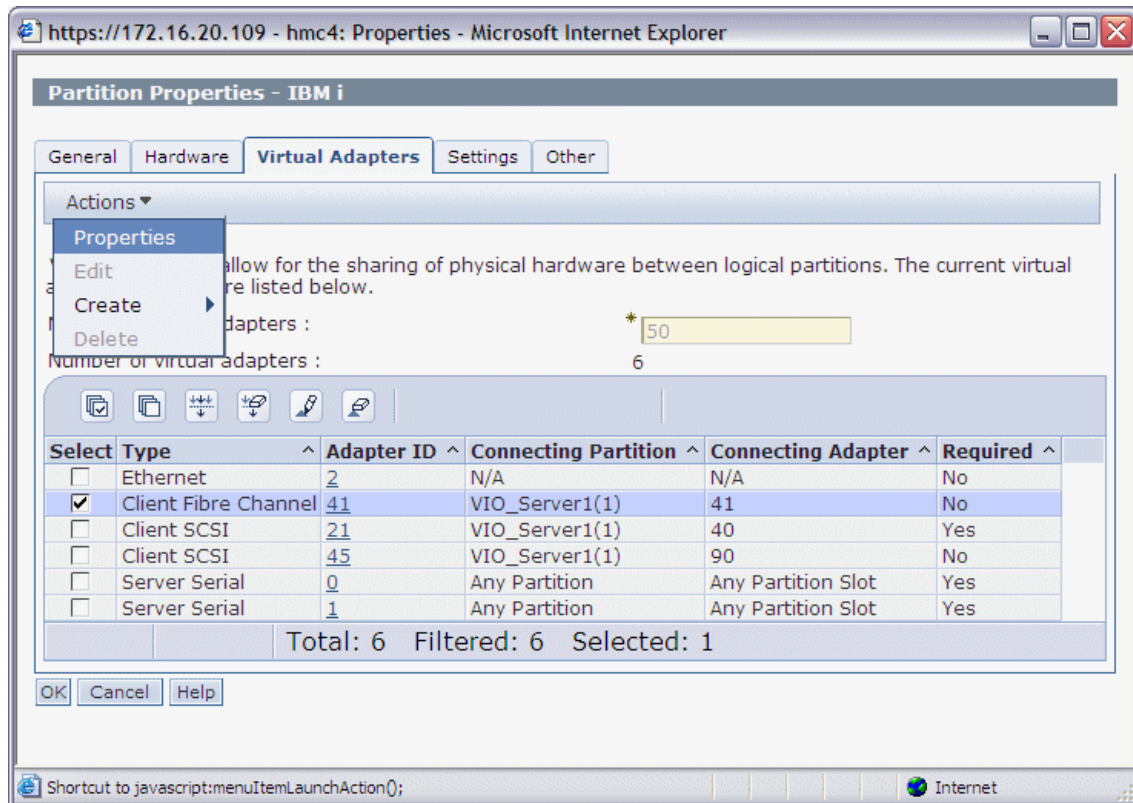


Figure 16-22 Select virtual Fibre Channel client adapter properties

- ii. Figure 16-23 shows the properties of the virtual Fibre Channel client adapter. Here you can get the virtual WWPN that is required for the zoning.

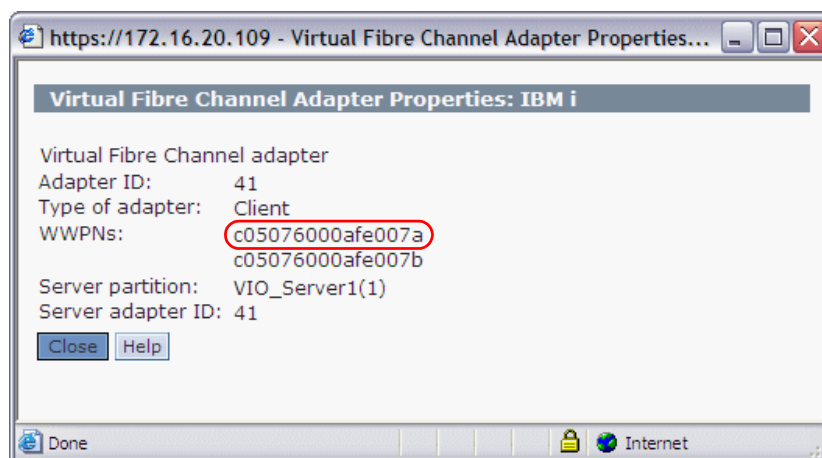


Figure 16-23 Virtual Fibre Channel client adapter properties

Tip: Unless using Live Partition Mobility or POWER7 partition suspend/resume, only the first listed WWPN is used and needs to be considered for the SAN zoning and storage configuration.

- b. Logon to your SAN switch and create a new zone for the virtual WWPN and the corresponding physical storage ports, or customize an existing one.

- c. After completing the SAN switch zoning, create the desired storage configuration on your SAN storage system with mapping the LUNs to a host connection created with the virtual WWPN of the virtual Fibre Channel client adapter. In our example we created four iSeries LUNs 1000, 1001, 1100, and 1101 on the DS8300, included them into a volume group and mapped them to the IBM i host connection as shown in Example 16-39.

Important for IBM i only: Because with Virtual Fibre Channel, in contrast to virtual SCSI, the storage LUNs are seen by the virtual I/O client partition with all their device characteristics as if they will be native-attached, the hostconnection and LUNs on the DS8000 are required to be created as iSeries host type and fixed size os400 volume types.

Example 16-39 DS8300 storage configuration for NPIV with IBM i

```
dsccli> mkfbvol -extpool P0 -os400 A02 -name IBMi_#h 1000-1001
Date/Time: 1. Dezember 2010 01:05:36 CET IBM DSCLI Version: 6.5.15.19 DS: IBM.2107-75BALB1
CMUC00025I mkfbvol: FB volume 1000 successfully created.
CMUC00025I mkfbvol: FB volume 1001 successfully created.
dsccli> mkfbvol -extpool P1 -os400 A02 -name IBMi_#h 1100-1101
Date/Time: 1. Dezember 2010 01:05:59 CET IBM DSCLI Version: 6.5.15.19 DS: IBM.2107-75BALB1
CMUC00025I mkfbvol: FB volume 1100 successfully created.
CMUC00025I mkfbvol: FB volume 1101 successfully created.
dsccli> mkvolgrp -type os400mask -volume 1000-1001 IBMi_01
Date/Time: 1. Dezember 2010 01:06:18 CET IBM DSCLI Version: 6.5.15.19 DS: IBM.2107-75BALB1
CMUC00030I mkvolgrp: Volume group V7 successfully created.
dsccli> chvolgrp -action add -volume 1100-1101 V7
Date/Time: 1. Dezember 2010 01:06:28 CET IBM DSCLI Version: 6.5.15.19 DS: IBM.2107-75BALB1
CMUC00031I chvolgrp: Volume group V7 successfully modified.
dsccli> mkhostconnect -wwname C05076000AFE007A -lbs 520 -profile "IBM iSeries - OS/400" -volgrp V7 IBMi
Date/Time: 1. Dezember 2010 01:06:51 CET IBM DSCLI Version: 6.5.15.19 DS: IBM.2107-75BALB1
CMUC00012I mkhostconnect: Host connection 000E successfully created.
```

- d. After completing the SAN storage configuration the volumes configured to the virtual Fibre Channel client adapter are now ready for use by the Virtual I/O Server client partition – for AIX if the virtual Fibre Channel devices were added dynamically run the **cfgmgr** command to scan for newly attached devices as shown in Example 16-40, on IBM i the new virtual Fibre Channel devices report in automatically (using the system value default QAUTOCFG=1) as shown in Figure 16-24.

Example 16-40 AIX Virtual Fibre Channel attached devices dynamic configuration and listing

```
# lsdev -Cc disk
hdisk0 Available Virtual SCSI Disk Drive
# cfgmgr
# lsdev -Cc disk
hdisk0 Available Virtual SCSI Disk Drive
hdisk1 Available 31-T1-01 IBM MPIO FC 2107
hdisk2 Available 31-T1-01 IBM MPIO FC 2107
```

Logical Hardware Resources Associated with IOP			
Type options, press Enter.			
2=Change detail 4=Remove 5=Display detail 6=I/O debug			
7=Verify 8=Associated packaging resource(s)			
Opt	Description	Type-Model	Status
	Virtual IOP	6B25-001	Operational
	Virtual Storage IOA	6B25-001	Operational
	Disk Unit	2107-A02	Operational
	Disk Unit	2107-A02	Operational
	Disk Unit	2107-A02	Operational
	Disk Unit	2107-A02	Operational
			Resource Name
			CMB07
			DC04
			DD005
			DD006
			DD007
			DD008
F3=Exit F5=Refresh F6=Print F8=Include non-reporting resources			
F9=Failed resources F10=Non-reporting resources			
F11=Display serial/part numbers F12=Cancel			

Figure 16-24 IBM i logical hardware resources with Virtual Fibre Channel devices

From the Linux client perspective, virtual Fibre Channel has to look like a native/physical Fibre Channel device. There is no special requirement or configuration needed to set up a Virtual Fibre Channel (VFC) on Linux.

After the `ibmvfc` driver is loaded and a virtual Fibre Channel Adapter is mapped to a physical Fibre Channel adapter on the Virtual I/O Server, the Fibre Channel port automatically shows up on the Linux partition. You can check if the `ibmvfc` driver is loaded on the system with the `lsmod` command:

```
[root@Power7-2-RHEL ~]# lsmod |grep ibmvfc
ibmvfc                98929  4
scsi_transport_fc     84177  1 ibmvfc
scsi_mod              245569  6
scsi_dh,sg,ibmvfc,scsi_transport_fc,ibmvscsic,sd_mod
```

You can also check the devices on the kernel log at the `/var/log/messages` file or by using the `dmesg` command output:

```
[root@Power7-2-RHEL ~]# dmesg |grep vfc
ibmvfc: IBM Virtual Fibre Channel Driver version: 1.0.6 (May 28, 2009)
vio_register_driver: driver ibmvfc registering
ibmvfc 30000038: Partner initialization complete
ibmvfc 30000038: Host partition: P7_2_vios1, device: vfchost0
U5802.001.0087356-P1-C2-T1 U8233.E8B.061AB2P-V1-C56 max sectors 2048
ibmvfc 30000039: Partner initialization complete
ibmvfc 30000039: Host partition: P7_2_vios2, device: vfchost0
U5802.001.0087356-P1-C3-T1 U8233.E8B.061AB2P-V2-C57 max sectors 2048
```

To list the virtual Fibre Channel device, use the command `lsscsi`, as shown in this example:

```
[root@Power7-2-RHEL ~]# lsscsi -H -v |grep fc
[5]    ibmvfc
[6]    ibmvfc
```

You can perform Virtual Fibre Channel tracing on Linux through the filesystem attributes located at the `/sys/class` directories. The files containing the devices' attributes are useful for checking detailed information about the virtual device and also can be used for troubleshooting as well. These attributes files can be accessed at the following directories:

```
/sys/class/fc_host/
/sys/class/fc_remote_port/
/sys/class/scsi_host/
```

16.2.3 Virtual optical

A DVD or CD device assigned to the Virtual I/O Server partition can be virtualized for shared use by the Virtual I/O Server's client partitions.

Figure 16-25 shows the Virtual I/O Server and client partition virtual SCSI setup for the shared optical device with the Virtual I/O Server owning the physical optical device and virtualizing it by its virtual SCSI server adapter in slot 90 configured for Any client partition can connect.

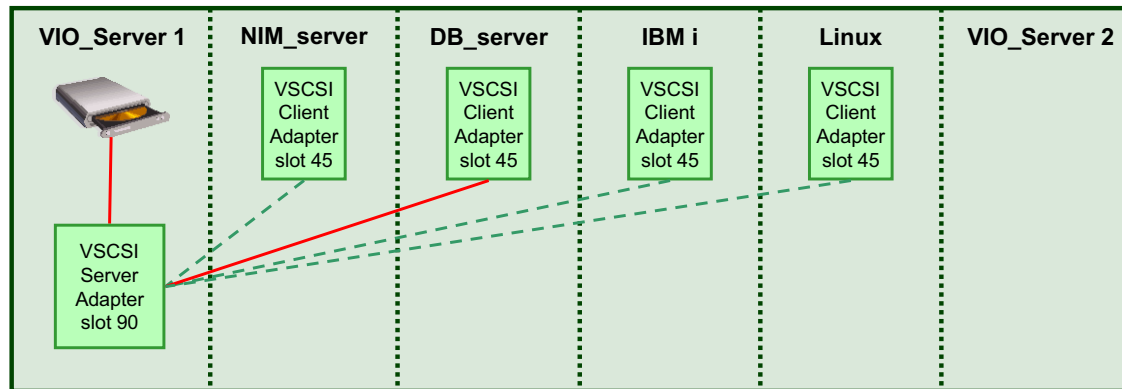


Figure 16-25 SCSI setup for shared optical device

Only one virtual I/O client partition can have access to the drive at a time like shown in Figure 16-25 for the DB_server partition currently accessing the virtualized optical device. The advantage of a virtual optical device is that you do not have to move the parent SCSI adapter between virtual I/O clients, which might even not be possible when this SCSI adapter also controls the internal disk drives on which the Virtual I/O Server was installed.

Attention:

The virtual drive cannot be moved to another Virtual I/O Server because client SCSI adapters cannot be created in a Virtual I/O Server. If you want the CD or DVD drive in another Virtual I/O Server, the virtual device must be unconfigured and the parent SCSI adapter must be unconfigured and moved using dynamic LPAR as described in *IBM PowerVM Virtualization Managing and Monitoring*, SG24-7590.

Creating a virtual optical device on the Virtual I/O Server

Follow these steps:

1. Assign the physical DVD drive to the Virtual I/O Server.
2. Create a virtual SCSI server adapter using the HMC where any partition can connect as shown in “Creating the Virtual I/O Server partition” on page 313.

Important: This must not be an adapter already used for disks because it will be removed or unconfigured when not holding the optical drive.

3. Run the **cfgdev** command to get the new vhost adapter. You can find the new adapter number with the **lsdev -virtual** command.
4. In the Virtual I/O Server, VIO_Server, you create the virtual device with the following command:

```
$ mkvdev -vdev <DVD drive> -vadapter vhostn -dev <any name>
```

Where n is the number of the vhost adapter. See Example 16-41.

Example 16-41 Making the virtual device for the DVD drive

```
$ mkvdev -vdev cd0 -vadapter vhost4 -dev vcd
```

5. Create a client SCSI adapter in each LPAR using the HMC. The client adapter must point to the server adapter created in the previous step. In our basic setup we used slot 90 for the server adapter and slot 45 for all client adapters.

Tip: Both virtual optical devices and virtual tape devices must be assigned dedicated virtual SCSI server-client adapter pairs. Because the server adapter is configured with the *Any client partition can connect* option, these pairs are not suited for client disks.

6. In the AIX client, run the **cfgmgr** command to assign the optical drive to the LPAR. If the drive is already assigned to another LPAR, you will receive an error message and will have to release the drive from the LPAR that is holding it.

On IBM i, the virtual optical device reports in automatically (when using the default system value QAUTOCFG=1) with a resource name of OPTxx and a device type of 632C-002 under a virtual SCSI IOP/IOA type 290A.

On Linux the partition needs to be rebooted to be able to assign the free drive. Likewise, the Linux partition needs to be shut down to release the drive. Linux automatically detects virtual optical devices provided by the Virtual I/O Server. Usually virtual optical devices are named as /dev/sr<ID>.

For further information on managing virtual optical devices on AIX, IBM i, and Linux like moving the shared device to another client partition or using it on the Virtual I/O Server itself refer to *IBM PowerVM Virtualization Managing and Monitoring*, SG24-7590.

16.2.4 Virtual tape

This sections describes the steps to set up a shared virtual tape drive which can be accessed by one Virtual I/O Server client partition at a time:

1. Assign the physical tape drive to the Virtual I/O Server partition.
2. Create a virtual SCSI server adapter using the HMC to which *any partition* can connect.

Important: Do not allow this adapter to be shared with disks, because it will be removed or unconfigured when not holding the tape drive.

3. Run the **cfgdev** command to configure the new vhost adapter. You can find the new adapter number with the **lsdev -virtual** command.
4. In the Virtual I/O Server, VIOS1, you create the virtual target device with the following command:

```
mkvdev -vdev tape_drive -vadapter vhostn -dev device_name
```

Where *n* is the number of the vhost adapter and *device_name* is the name for the virtual target device. See Example 16-42.

Example 16-42 Making the virtual device for the tape drive

```
$ mkvdev -vdev rmt0 -vadapter vhost3 -dev vtape
```

5. Create a virtual SCSI client adapter in each LPAR using the HMC. The client adapter must point to the server adapter created in Step 4. In the scenario, slot 60 is used on the server and also for each of the client adapters.

Tip: It is useful to use the same slot number for all the clients.

6. In the AIX client, run the **cfgmgr** command to assign the drive to the LPAR. If the drive is already assigned to another LPAR, you will receive an error message and will have to release the drive from the LPAR that is holding it.

On IBM i, the virtual tape device reports in automatically (when using the default system value QAUTOCFG=1) with a resource name of TAPxx with its corresponding physical device type such as 3580-004 for a SAS LTO4 tape under a virtual SCSI IOP/IOA type 290A.

On Linux, the virtual tape device is automatically detected when the virtual tape is created on the Virtual I/O Server. Virtual tape devices are named just like physical tapes, such as /dev/st0, /dev/st1, and so on. Use the **mt** command for managing and tracing tape devices on Linux.

For further information on managing virtual tape devices on AIX, IBM i, and Linux like moving the shared device to another client partition refer to *IBM PowerVM Virtualization Managing and Monitoring*, SG24-7590.

16.2.5 Availability

This section talks about details on setup required to configure availability of virtual storage at client side. We assume you are already familiar with overview and planning parts of this book.

Multipathing in the client partition

With multipathing in the virtual I/O client partition, each Virtual I/O Server can present a virtual SCSI or virtual Fibre Channel device that is physically connected to the same physical disk. This achieves redundancy for the Virtual I/O Server itself and for any adapter, switch, or device that is used between the Virtual I/O Server and the disk.

The following sections describe using multipathing for each different AIX, IBM i and Linux client partition across two Virtual I/O Servers as shown in Figure 16-26 on page 497.

Multipathing in the AIX client partition

MPIO for virtual Fibre Channel devices in the AIX client partition does not require any specific configuration. Virtual Fibre Channel in the client partition allows the use of the existing tools and techniques for storage management, including multipathing software. MPIO for virtual Fibre Channel devices in the AIX client partition supports round robin, load balancing, and failover mode.

MPIO for virtual SCSI devices in the AIX client partition only supports failover mode. For any given virtual SCSI disk, a client partition will use a primary path to one Virtual I/O Server and fail over to the secondary path to use the other Virtual I/O Server. Only one path is used at a given time even when both paths are enabled.

To balance the load of multiple client partitions across two Virtual I/O Servers, the priority on each virtual SCSI disk on the client partition can be set to select the primary path and, therefore, a specific Virtual I/O Server. The priority is set on a per virtual SCSI disk basis using the **chpath** command as shown in the following example (1 is the highest priority):

```
chpath -l hdisk0 -p vscsi0 -a priority=2
```

Due to this granularity, a system administrator can specify whether all the disks or alternate disks on a client partition use one of the Virtual I/O Servers as the primary path. The best method is to divide the client partitions between the two Virtual I/O Servers.

Important: The priority path can be set per VSCSI LUN, enabling fine granularity of load balancing in dual Virtual I/O Server configurations between primary and backup.

For MPIO support of virtual SCSI devices in the AIX client partition, the SAN LUN must be presented as a physical drive (hdiskx) from the Virtual I/O Server to the client partition. It is not possible to provide a large SAN LUN and then further subdivide it into logical volumes at the Virtual I/O Server level when using two Virtual I/O Servers. There is no volume group created on the SAN LUNs on the Virtual I/O Server. The storage management for this configuration is performed in the SAN, so there is a one-to-one mapping of SAN LUNs on the Virtual I/O Servers to virtual SCSI drives on the client partition.

Important: If each Virtual I/O Server has a different number of drives or the drives were zoned at different times, the device names (hdiskx) might be different between Virtual I/O Servers. Always check that the LUN IDs match when presenting a drive to the same client partition using dual Virtual I/O Servers. It is useful from an administration point of view to have the same device names on both Virtual I/O Servers.

The heartbeat check interval for each disk using MPIO must be configured so the path status is updated automatically. Specifying *hcheck_mode=nonactive* means that healthcheck commands are sent down paths that have no active I/O, including paths with a state of failed. The *hcheck_interval* attribute defines how often the healthcheck is performed. In the client partition the *hcheck_interval* for virtual SCSI devices is set to 0 by default which means healthchecking is disabled.

It must be enabled using the **chdev** command as shown in the following example:

```
chdev -l hdisk0 -a hcheck_interval=60 -a hcheck_mode=nonactive -P
```

Attention: Because the attribute cannot be changed while the vscsi device is in the active state, the **-P** flag is used so that the change is made in the ODM only. The changes are applied to the device when the system is rebooted.

Never set the `hcheck_interval` lower than the read/write timeout value of the underlying physical disk on the Virtual I/O Server. Otherwise, an error detected by the Fibre Channel adapter causes new healthcheck requests to be sent before the running requests time out. The consequence is a backlog of normal I/O requests from applications or databases waiting on path health checks to complete. In the event of adapter or path issues, setting the `hcheck_interval` too low can cause severe performance degradation or possibly cause I/O hangs.

The minimum recommended value for the `hcheck_interval` attribute is 60 for both Virtual I/O and non-Virtual I/O configurations.

Example 16-43 shows a properly configured `hdisk` device on a Virtual I/O Server.

Example 16-43 Properly configured `hcheck_interval`

\$ lsdev -dev hdisk30 -attr			
attribute	value	description	user_settable
PCM	PCM/friend/sddpcm	PCM	True
PR_key_value	none	Reserve Key	True
algorithm	load_balance	Algorithm	True
clr_q	no	Device CLEARS its Queue on error	True
dist_err_pcmt	0	Distributed Error Percentage	True
dist_tw_width	50	Distributed Error Sample Time	True
hcheck_interval	60	Health Check Interval	True
hcheck_mode	nonactive	Health Check Mode	True
location		Location Label	True
lun_id	0x70000000000000	Logical Unit Number ID	False
lun_reset_spt	yes	Support SCSI LUN reset	True
max_transfer	0x40000	Maximum TRANSFER Size	True
node_name	0x5005076801003701	FC Node Name	False
pvid	00c1ea606260227c0000000000000000	Physical volume identifier	False
q_err	yes	Use QERR bit	True
q_type	simple	Queueing TYPE	True
qfull_dly	20	delay in seconds for SCSI TASK SET FULL	True
queue_depth	20	Queue DEPTH	True
reserve_policy	no_reserve	Reserve Policy	True
rw_timeout	60	READ/WRITE time out value	True
scbsy_dly	20	delay in seconds for SCSI BUSY	True
scsi_id	0x641413	SCSI ID	False
start_timeout	180	START unit time out value	True
unique_id	332136005076801918129480000000000031904214503IBMfc	Device Unique Identification	False
ww_name	0x5005076801403701	FC World Wide Name	False
\$			

Tip: To further limit the number of healthcheck commands, it is best not to configure more than 4 to 8 paths per LUN, and set the `hcheck_interval` to 60 in the client partition and on the Virtual I/O Server.

Important: When using an SSDPCM level higher than 2.1.2.3, the default value for the `hcheck_interval` is 60 for the `hdisk` devices on the Virtual I/O Server. If you are using a lower level of SSDPCM, you need to increase the `hcheck_interval` level manually for each `hdisk` device.

The *queue depth* value for each disk using MPIO on the client partition, which determines how many requests the disk head driver will queue to the virtual SCSI client driver at any one time, must be configured to match the queue depth value used for the physical disk on the Virtual I/O Server. It must be changed using the **chdev** command as shown in the following example:

```
chdev -l hdisk0 -a queue_depth=20 -P
```

On the virtual SCSI client adapter, the virtual SCSI adapter *path timeout* must be configured. It allows the client adapter to check the health of the Virtual I/O Server servicing a particular adapter and detect if a Virtual I/O Server is not responding to I/O requests. In such a case, the client will failover to an alternate path if the Virtual SCSI adapter path timeout is configured. This path timeout is configured using the **chdev** command as shown in the following example.

```
chdev -l vscsi0 -a vscsi_path_to=30 -P
```

Starting with AIX 5.3 TL9 (APAR IZ28537) and AIX 6.1 TL2 (APAR IZ28554), a new Virtual SCSI adapter *error recovery* parameter has been added and must be configured to `fast_fail`. When this parameter is set to `fast_fail`, the virtual client adapter sends `FAST_FAIL_MAD` to the Virtual I/O Server and it fails the I/Os immediately rather than delayed. This parameter is configured using the **chdev** command as shown in the following example.

```
chdev -l vscsi0 -a vscsi_err_recov=fast_fail -P
```


Figure 16-26 shows a configuration where MPIO is used in the client partition. The Virtual I/O Server is using SAN storage, provided IBM System Storage SAN Volume Controller, and the SDDPCM module. It shows which attributes must be set in this configuration. MPIO in the client partition can be used without configuring MPIO on the Virtual I/O Server. This will be the case if each Virtual I/O Server has just a single Fibre Channel adapter configured.

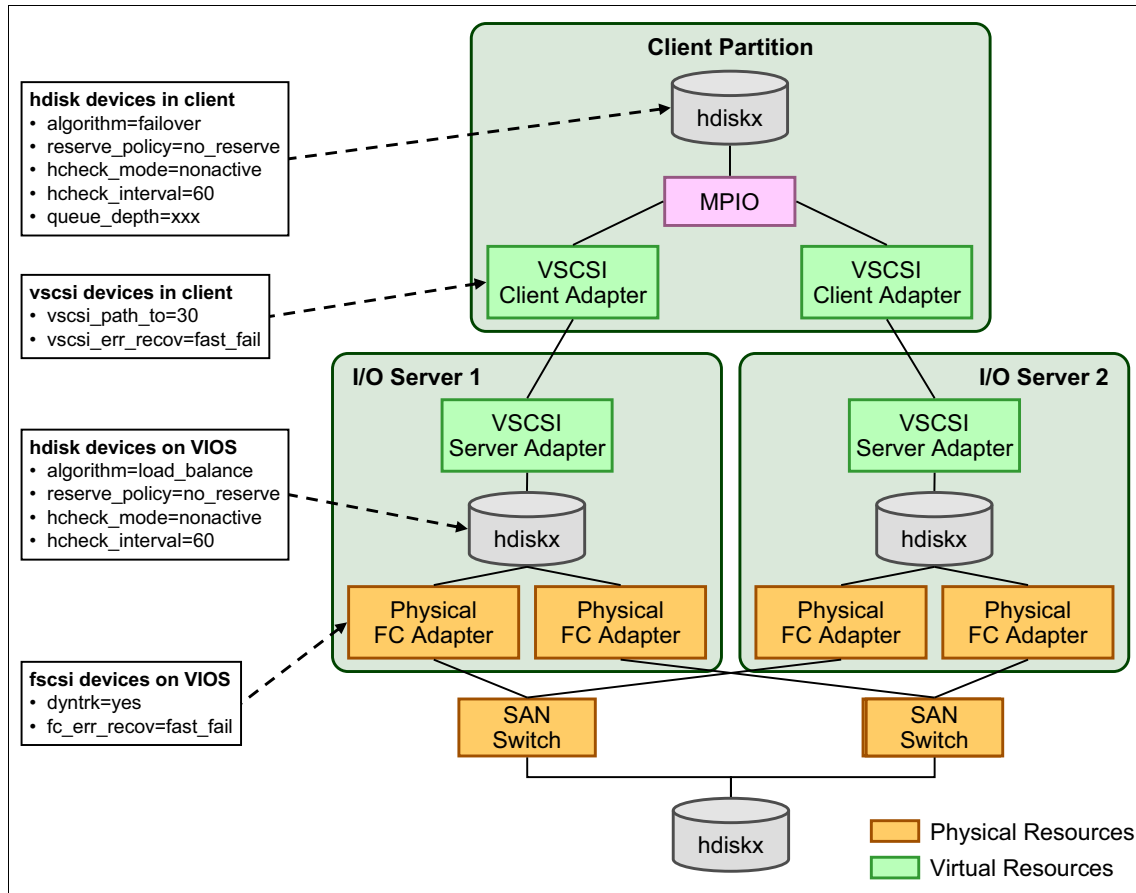


Figure 16-26 MPIO attributes

The settings when using multiple Fibre Channel adapters and MPIO on the Virtual I/O Server level are described in detail in “Availability configurations using multipathing” on page 501.

Multipathing in the IBM i client partition

There is nothing special to consider for using IBM i multipathing with virtual SCSI or virtual Fibre Channel devices, which is supported with IBM i 6.1.1 or later, compared to physical devices. In the following topics we provide background information about using IBM i multipathing.

Support: IBM i multipathing is not supported for tape devices.

The same multipathing algorithm and rules apply as for native-attached storage systems, that is, IBM i multipathing uses a round-robin algorithm for load balancing and supports up to 8 paths supported per disk unit.

After a disk unit is recognized by IBM i as a multipath disk unit, that is, the disk unit enlists by a second path, its resource name gets automatically changed from DDxxx to DMPxxx. However, if for any reason, paths fail or get removed again such that only one operational path is left, the resource name is not changed back to DDxxx. If paths have intentionally been removed by a configuration change, they must be removed from the IBM i configuration *after* an IPL, to clean up orphan resources and prevent delays at further IPLs for the System Licensed Internal Code waiting for the missing paths to show up, by using the System Service Tools's MULTIPATHRESETTER macro or the corresponding function in the Hardware Service Manager (for further information, see *IBM i and IBM System Storage: A Guide to Implementing External Disks on IBM i*, SG24-7120).

Paths to a multipath disk unit that failed and become operational again are automatically used again by IBM i without user intervention.

See “IBM i client multipathing” on page 517 for an example of an IBM i multipathing configuration.

Multipathing in the Linux client partition

Multipathing using virtual I/O on Linux can be done through virtual SCSI adapters or virtual Fibre Channel adapters. Although these multipathing solutions are similar, the former solution uses *ibmvscsi* driver and the disk is virtualized. The latter uses *ibmvfc* driver and the Fibre Channel is virtualized. See “Virtual Fibre Channel” on page 474 an example of Virtual Fibre Channel configuration for multipathing.

Linux automatically detects disks as long as their adapters are correctly mapped in the partition profile. After the physical devices are assigned to the Virtual I/O Servers and mapped to the client partition, you can list the devices with **fdisk** or **ls SCSI** commands. Device paths are named as regular physical disks such as /dev/sda or /dev/sdb.

The module *dm_multipath* must be loaded to detect multipath devices in the system. The round-robin algorithm is used for load balancing, making the I/O operations to be split among the paths. Multipath devices are named as mpath0, mpath1, and so on.

Attention: Naming conventions for multipath devices can vary between one Linux distribution and another.

Although there are no special considerations to implement a multipath solution on Linux, certain requirements must be observed:

- ▶ The *ibmvscsi* driver must be loaded.
- ▶ The *ibmvfc* driver must be loaded when using Virtual Fibre Channel.
- ▶ The *dm_multipath* module must be loaded.
- ▶ The */etc/multipath.conf* must be edited accordingly.
- ▶ The *multipathd* daemon must be started.
- ▶ The **multipath** command must be used for tracing.

To install Linux on a multipath device, you must ensure that dm-multipath module is loaded during the installation and specify the multipath device accordingly.

Failed paths are automatically detected by Linux, and recovered paths become active again without user intervention. These events are logged and can be found at the */var/log/messages* file.

See “Linux client multipathing” on page 529 for an example of a Linux multipathing configuration.

Multipathing in the Virtual I/O Server

This section describes the configuration of the Virtual I/O Server when deploying MPIO in a dual Virtual I/O Servers environment.

Fibre Channel device configuration

The fscsi devices include specific attributes that must be changed on both Virtual I/O Servers. These attributes are *fc_err_recov* and the *dyntrk* attribute. Both attributes can be changed using the **chdev** command as follows:

```
$ chdev -dev fscsi0 -attr fc_err_recov=fast_fail dyntrk=yes -perm  
fscsi0 changed
```

Changing the `fc_err_recov` attribute to `fast_fail` will fail any new or retried I/Os immediately after the adapter detects a link event, such as a lost link between a storage device and a switch. The `fast_fail` setting is only desired when using multipathing in the Virtual I/O Server. Setting the `dyntrk` attribute to `yes` allows the Virtual I/O Server to tolerate cabling changes in the SAN. Both Virtual I/O Servers need to be rebooted for these changed attributes to take effect.

hdisk device configuration on the Virtual I/O Server

To correctly enable the presentation of a physical drive to a client partition by dual Virtual I/O Servers, the *reserve_policy* attribute on each disk must be set to `no_reserve`. Using `hdisk1` as an example, use the `chdev` command to change both the reserve policy and algorithm on the `hdisk`:

```
$ chdev -dev hdisk1 -attr reserve_policy=no_reserve
hdisk1 changed
```

In addition, to enable load balancing across multiple Fibre Channel adapters within the Virtual I/O Servers when using the base AIX MPIO support, change the default *fail_over* algorithm to *round_robin* for each physical disk as shown in Example 16-44. This is not required when you use SDD or SDDPCM as multipathing software:

Example 16-44 Changing disks to round_robin algorithm

```
$ chdev -dev hdisk1 -attr algorithm=round_robin
hdisk1 changed
$ lsdev -dev hdisk1 -attr
```

attribute	value	description	
user_settable			
PCM	PCM/friend/scsiscsd	Path Control Module	False
algorithm	round_robin	Algorithm	True
dist_err_pcnt	0	Distributed Error Percentage	True
dist_tw_width	50	Distributed Error Sample Time	True
hcheck_interval	0	Health Check Interval	True
hcheck_mode	nonactive	Health Check Mode	True
max_transfer	0x40000	Maximum TRANSFER Size	True
pvid	0021768a0151feb40000000000000000	Physical volume identifier	False
queue_depth	3	Queue DEPTH	False
reserve_policy	no_reserve	Reserve Policy	True
size_in_mb	18200	Size in Megabytes	False

16.2.6 Availability configurations using multipathing

The scenarios in this section show how to set up a highly available virtual SCSI or virtual Fibre Channel (VFC) configuration using multipathing in the client partition. Figure 16-27 on page 502 shows the configuration we use for the scenarios of implementing multipathing in AIX, IBM i and Linux client partitions.

The disks are located on an external storage subsystem on the SAN. For virtual SCSI the disks on the storage subsystem are assigned to both Virtual I/O Servers, while for virtual Fibre Channel each client has the disks assigned to both of its virtual Fibre Channel adapters. The client partitions see the disks through two paths using multipathing. Each of the paths is going through a different Virtual I/O Server.

Support: For AIX client partitions, MPIO for virtual SCSI devices only supports failover mode.

In our scenario, the disk subsystem is an IBM System Storage DS8000. To provide Fibre Channel adapter redundancy each Virtual I/O Server is attached to the DS8000 SAN storage using two Fibre Channel adapters. For the virtual SCSI scenarios, besides the required DS8000 Host Attachment script, the Subsystem Device Driver Path Control Module (SDDPCM) is installed for multipath access of the Virtual I/O Server itself to the DS8000 SAN storage.

Figure 16-27 depicts attachment with multipathing across two Virtual I/O Servers.

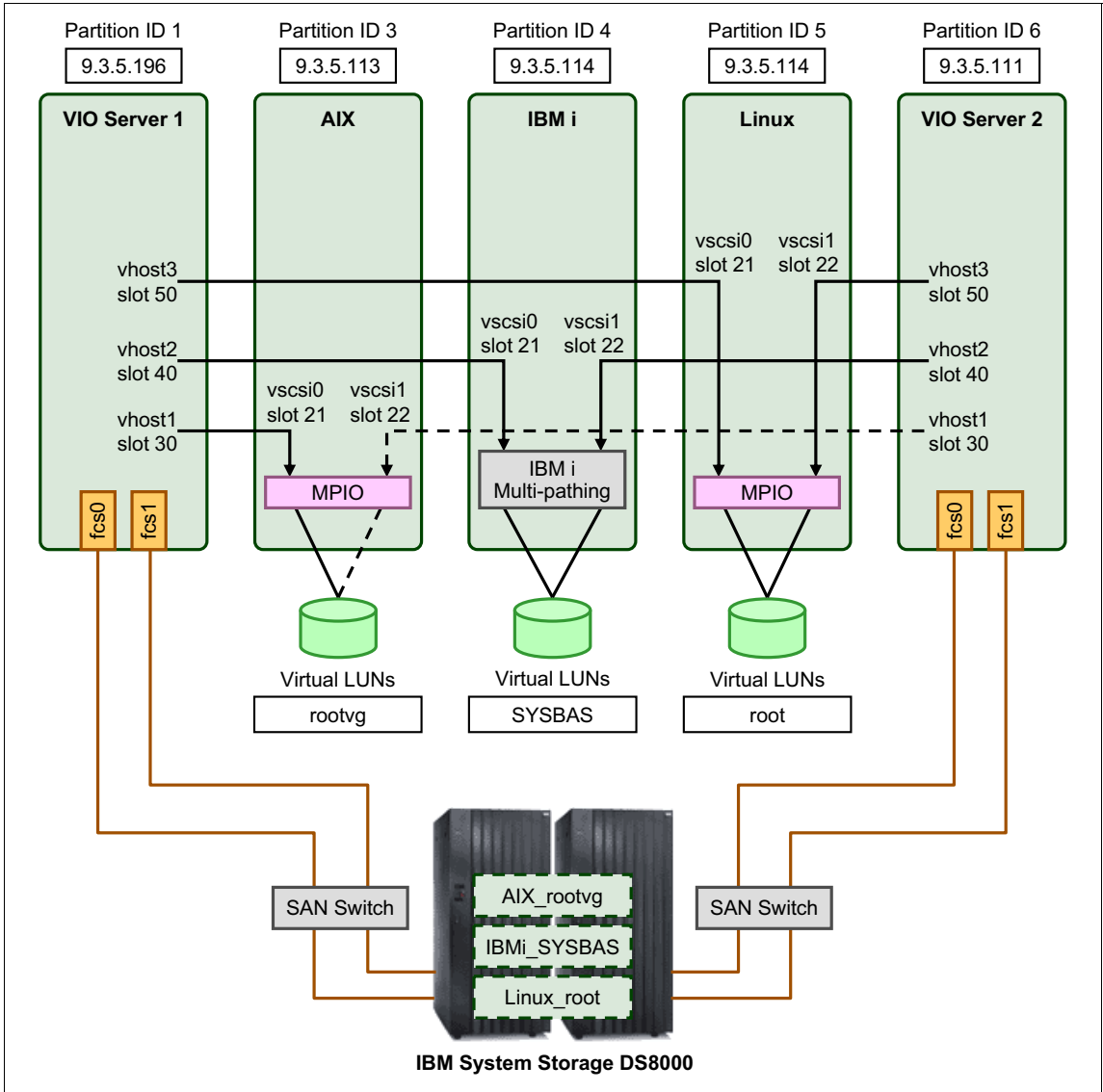


Figure 16-27 SAN attachment with multipathing across two Virtual I/O Servers

When using midrange storage systems such as IBM System Storage DS4000 or DS5000 series, which process I/O for a particular storage LUN only by one of its two storage controllers and using only one Fibre Channel adapter port per Virtual I/O Server, you need an additional switch to have a highly available configuration. In this case, it is important that the SAN zoning be configured such that the single Fibre Channel adapter port in each Virtual I/O Server partition is zoned to see both storage controllers of the midrange storage system.

Configuring multipathing in the server

Use the following steps to set up the scenario:

1. Create two Virtual I/O Server partitions and name them VIO_Server1 and VIO_Server2, following the instructions in 12.1, “Creating a Virtual I/O Server” on page 312. In step 10, select one Fibre Channel adapter in addition to the shown physical adapter.
2. Install both Virtual I/O Servers by following the instructions in 12.2, “Installation of Virtual I/O Server” on page 333
3. Change the *fc_err_recov* to *fast_fail* and *dyntrk* to *yes* attributes on the Fibre Channel adapters’ SCSI protocol devices *fscsiX*. You can use the **lsdev | grep fscsi** command to list them which are as many as there are physical Fibre Channel adapter ports. Use the **chdev** command, as shown in Example 16-45, to change the attributes.

Example 16-45 Changing the attributes of the Fibre Channel adapter

```
$ chdev -dev fscsi0 -attr fc_err_recov=fast_fail dyntrk=yes -perm
fscsi0 changed
$ lsdev -dev fscsi0 -attr
```

attribute	value	description	user_settable
attach	switch	How this adapter is CONNECTED	False
dyntrk	yes	Dynamic Tracking of FC Devices	True
fc_err_recov	fast_fail	FC Fabric Event Error RECOVERY Policy	True
scsi_id	0x660c00	Adapter SCSI ID	False
sw_fc_class	3	FC Class for Fabric	True

The reason for changing the *fc_err_recov* attribute to *fast_fail* is that if the Fibre Channel adapter driver detects a link event, such as a lost link between a storage device and a switch, then any new I/O or future retries of the failed I/O operations will be failed immediately by the adapter until the adapter driver detects that the device has rejoined the fabric. The default setting for this attribute is *delayed_fail*.

Setting the *dyntrk* attribute to *yes* makes AIX tolerate cabling changes in the SAN. Be aware that this function is not supported on all storage systems. Check with your storage vendor for support.

Important: If you have two or more Fibre Channel adapters per Virtual I/O Server you have to change the attributes for each of them.

4. Reboot the Virtual I/O Servers for the changes to the Fibre Channel devices to take effect.
5. Create the client partitions following the instructions in “Creating a client partition” on page 354. Table 16-6 shows the required virtual SCSI adapters based on the configuration shown in Table 16-6.

Table 16-6 Virtual SCSI adapter configuration for MPIO

Virtual I/O Server	Virtual I/O Server slot	Client partition	Client partition slot
VIO_Server1	30	AIX	21
VIO_Server1	40	IBM i	21
VIO_Server1	50	Linux	21
VIO_Server2	30	AIX	22
VIO_Server2	40	IBM i	22
VIO_Server2	50	Linux	22

6. Also add one or two virtual Ethernet adapters to each client to provide the highly available network access: One adapter if you plan on using SEA failover for network redundancy, as described in “Configuring SEA failover” on page 593 or two adapters for AIX or Linux if you plan on using Network Interface Backup for network redundancy, as described in 16.3.3, “EtherChannel Backup in the AIX client” on page 603.

7. On VIO_Server1 and VIO_Server2 use the **pcmpath query device** command from `oem_setup_env` to get the LUN to hdisk mappings, as shown in Example 16-46. In this example, hdisk2 and hdisk3 are used for the client partitions. It is important that you identify the same disks on both Virtual I/O Servers.

Tip: Depending on the multipathing software you use on the Virtual I/O Server, the commands to identify the SAN disks will be different:

- ▶ If you use MPIO, use the **mpio_get_config -Av** command.
- ▶ If you use RDAC, use the **fget_config -Av** command.
- ▶ If you have SDDPCM installed, use **pcmpath query device** command.
- ▶ If you have SDD installed, use the **datapath query device** command.

Example 16-46 Listing the LUN to hdisk mappings

```
# pcmpath query device

Total Dual Active and Active/Asymmetric Devices : 20

DEV#: 3  DEVICE NAME: hdisk3  TYPE: 2107900  ALGORITHM: Load Balance
SERIAL: 75BALB11011
=====
Path#      Adapter/Path Name      State   Mode    Select   Errors
  0         fscsil/path1         CLOSE  NORMAL    20        0
  1         fscsi0/path0        CLOSE  NORMAL    17        0

DEV#: 4  DEVICE NAME: hdisk4  TYPE: 2107900  ALGORITHM: Load Balance
SERIAL: 75BALB11012
=====
Path#      Adapter/Path Name      State   Mode    Select   Errors
  0         fscsil/path1         OPEN   NORMAL    42        0
  1         fscsi0/path0        OPEN   NORMAL    25        0
...
DEV#: 10  DEVICE NAME: hdisk10  TYPE: 2107900  ALGORITHM: Load Balance
SERIAL: 75BALB11018
=====
Path#      Adapter/Path Name      State   Mode    Select   Errors
  0         fscsi0/path0        OPEN   NORMAL   369        0
  1         fscsil/path1        OPEN   NORMAL   303        0
...

```

You can also use the `lsdev -dev hdiskn -vpd` command, where *n* is the hdisk number, to retrieve this information, as shown in Example 16-47.

Example 16-47 Listing the LUN to hdisk mapping using the lsdev command

```
$ lsdev -dev hdisk3 -vpd
hdisk3
U5802.001.0086848-P1-C2-T1-W500507630410412C-L4010401100000000 IBM MPI0 FC
2107
```

```
Manufacturer.....IBM
Machine Type and Model.....2107900
Serial Number.....75BALB11011
EC Level.....278
Device Specific.(Z0).....10
Device Specific.(Z1).....0312
Device Specific.(Z2).....075
Device Specific.(Z3).....29205
Device Specific.(Z4).....08
Device Specific.(Z5).....00
```

PLATFORM SPECIFIC

```
Name: disk
Node: disk
Device Type: block
```

Tip: If possible, keep the hdisk numbering on the two Virtual I/O Servers identical. Having identical numbering makes the management of the mappings easier.

8. The disks are to be accessed through both Virtual I/O Servers. The *reserve_policy* for each disk must be set to *no_reserve* on VIO_Server1 and VIO_Server2. Change the *reserve_policy* attribute to *no_reserve* using the `chdev` command, as shown in Example 16-48.

Example 16-48 Set the attribute to no_reserve

```
$ chdev -dev hdisk3 -attr reserve_policy=no_reserve
hdisk3 changed
$ chdev -dev hdisk4 -attr reserve_policy=no_reserve
hdisk4 changed
```

9. Check, using the **lsdev** command, to make sure `reserve_policy` attribute is now set to `no_reserve`, as shown in Example 16-49.

Example 16-49 Output of the `hdisk` attribute with changed `reserve_policy` attribute

\$ lsdev -dev hdisk3 -attr			
attribute	value	description	
user_settable			
PCM	PCM/friend/sddpcm	PCM	True
PR_key_value	none	Reserve Key	True
algorithm	load_balance	Algorithm	True
clr_q	no	Device CLEARS its Queue on error	True
dist_err_pcmt	0	Distributed Error Percentage	True
dist_tw_width	50	Distributed Error Sample Time	True
hcheck_interval	60	Health Check Interval	True
hcheck_mode	nonactive	Health Check Mode	True
location		Location Label	True
lun_id	0x4010401100000000	Logical Unit Number ID	False
lun_reset_spt	yes	Support SCSI LUN reset	True
max_transfer	0x100000	Maximum TRANSFER Size	True
node_name	0x5005076304ffc12c	FC Node Name	False
pvid	none	Physical volume identifier	False
q_err	yes	Use QERR bit	True
q_type	simple	Queuing TYPE	True
qfull_dly	2	delay in seconds for SCSI TASK SET FULL	True
queue_depth	20	Queue DEPTH	True
reserve_policy	no_reserve	Reserve Policy	True
retry_timeout	120	Retry Timeout	True
rw_timeout	60	READ/WRITE time out value	True
scbsy_dly	20	delay in seconds for SCSI BUSY	True
scsi_id	0x10c00	SCSI ID	False
start_timeout	180	START unit time out value	True
unique_id	200B75BALB1101107210790003IBMfc	Device Unique Identification	False
ww_name	0x500507630410412c	FC World Wide Name	False

Important: The `no_reserve` policy has to be set for each `hdisk` that will be used for client multipathing across multiple Virtual I/O Servers.

10. Double-check both Virtual I/O Servers to make sure that the vhost adapters have the correct slot numbers by running the **lsmap -all** command.
11. Map the hdisks to the vhost adapters using the **mkvdev** command, as shown in Example 16-50.

Example 16-50 Mapping the disks to the vhost adapters

```
$ mkvdev -vdev hdisk4 -vadapter vhost0 -dev AIX_rootvg
AIX_rootvg Available
$ mkvdev -vdev hdisk10 -vadapter vhost5 -dev vIBMi_LS
vIBMi_LS Available
```

Important: When multiple Virtual I/O Servers attach to the same disk, only hdisk is supported as a backing device. You cannot create a volume group on these disks and use a logical volume as a backing device.

Check the mappings, as shown in Example 16-51.

Example 16-51 lsmap -all output after mapping the disks on VIO_Server1

```
$ lsmap -all
SVSA                Physloc                Client Partition ID
-----
vhost0              U8233.E8B.061AA6P-V1-C30 0x00000003

VTD                  AIX_rootvg
Status                Available
LUN                   0x8100000000000000
Backing device        hdisk4
Physloc
U5802.001.0086848-P1-C2-T1-W500507630410412C-L4010401200000000
Mirrored              false
...
SVSA                Physloc                Client Partition ID
-----
vhost5              U8233.E8B.061AA6P-V1-C40 0x00000004

VTD                  vIBMi_2
Status                Available
LUN                   0x8200000000000000
Backing device        hdisk11
Physloc
U5802.001.0086848-P1-C2-T1-W500507630410412C-L4010401900000000
Mirrored              false

VTD                  vIBMi_3
Status                Available
LUN                   0x8300000000000000
```

```

Backing device      hdisk12
Physloc
U5802.001.0086848-P1-C2-T1-W500507630410412C-L4010401A00000000
Mirrored           false

VTD                 vIBMi_4
Status              Available
LUN                 0x8400000000000000
Backing device      hdisk13
Physloc
U5802.001.0086848-P1-C2-T1-W500507630410412C-L4011401000000000
Mirrored           false

VTD                 vIBMi_LS
Status              Available
LUN                 0x8100000000000000
Backing device      hdisk10
Physloc
U5802.001.0086848-P1-C2-T1-W500507630410412C-L40104018000000000
Mirrored           false

```

Notice the highlighted DS8000 volume IDs included in the Physloc output from the lsmmap listing from both Virtual I/O Servers which need to match for the mapping of virtual SCSI devices for each client partition as shown in Example 16-51 on page 508 and Example 16-52. This allows the virtual I/O client partitions to see each of their virtual SCSI LUNs, backed by the same DS8000 volume, by two Virtual I/O Servers.

Example 16-52 lsmmap -all output after mapping the disks on VIO_Server2

```

$ lsmmap -all
SVSA              Physloc                      Client Partition ID
-----
vhost0            U8233.E8B.061AA6P-V2-C30 0x00000003

VTD               AIX_rootvg
Status            Available
LUN               0x8100000000000000
Backing device    hdisk4
Physloc
U5802.001.0086848-P1-C3-T1-W500507630414C12C-L40104012000000000
Mirrored         false
...

```

SVSA	Physloc	Client Partition ID
vhost3	U8233.E8B.061AA6P-V2-C40	0x00000006
VTD	vIBMi_2	
Status	Available	
LUN	0x8200000000000000	
Backing device	hdisk11	
Physloc	U5802.001.0086848-P1-C3-T1-W500507630414C12C-L4010401900000000	
Mirrored	false	
VTD	vIBMi_3	
Status	Available	
LUN	0x8300000000000000	
Backing device	hdisk12	
Physloc	U5802.001.0086848-P1-C3-T1-W500507630414C12C-L4010401A00000000	
Mirrored	false	
VTD	vIBMi_4	
Status	Available	
LUN	0x8400000000000000	
Backing device	hdisk13	
Physloc	U5802.001.0086848-P1-C3-T1-W500507630414C12C-L4011401000000000	
Mirrored	false	
VTD	vIBMi_LS	
Status	Available	
LUN	0x8100000000000000	
Backing device	hdisk10	
Physloc	U5802.001.0086848-P1-C3-T1-W500507630414C12C-L4010401800000000	
Mirrored	false	

12. Install the client partitions. For further information about installing an AIX or IBM i client partition, see the corresponding section of “Creating a client partition” on page 354.

AIX client multipathing

The following steps are used to configure MPIO in the AIX client partitions.

Overview

In our example the DB_server partition uses VIO_Server1 as the primary path for virtual SCSI traffic. The APP_server partition will use VIO_Server2 as the primary path.

Support: MPIO for virtual SCSI devices currently only support failover mode in AIX.

Alternatively, if you use SEA failover for network redundancy, you can direct all virtual SCSI traffic to the backup Virtual I/O Server for the network. See 10.1.2, “Redundancy considerations” on page 175 for details on separating traffic.

The following steps show how to configure the client partitions:

1. Check the MPIO configuration by running the commands shown in Example 16-53. Only one configured hdisk shows up in this scenario.

Example 16-53 Verifying the disk configuration on the client partitions

```
# lspv
hdisk0          00c1f1707355c8e5          rootvg
active
# lsdev -Cc disk
hdisk0 Available Virtual SCSI Disk Drive
```

2. Run the **lspath** command to verify that the disk is attached using two different paths. Example 16-54 shows that hdisk0 is attached using the VSCSI0 and VSCSI1 adapter that point to different Virtual I/O Servers. Both Virtual I/O Servers are up and running. Both paths are enabled.

Example 16-54 Verifying the paths of hdisk0

```
# lspath
Enabled hdisk0 vscsi0
Enabled hdisk0 vscsi1
```

3. Enable the health check mode for the disk so that the status of the disks is automatically updated. Health check mode is disabled by default (hcheck_interval=0). As long as it is disabled the client partition does not update the path status in case of a failure of the active path. To activate the health check function, use the **chdev** command, as shown in Example 16-55. In this example, we use a health check interval of 50 seconds. To check for the attribute setting, use the **lsattr** command.

Tip: The path switching also works if the hcheck_interval attribute is disabled, but it still has to be set to have the status updated automatically.

Example 16-55 Changing the health check interval

```
# chdev -l hdisk0 -a hcheck_interval=50 -P
hdisk0 changed
# lsattr -El hdisk0
```

PCM	PCM/friend/vscsi	Path Control Module	False
algorithm	fail_over	Algorithm	True
hcheck_cmd	test_unit_rdy	Health Check Command	True
hcheck_interval	50	Health Check Interval	True
hcheck_mode	nonactive	Health Check Mode	True
max_transfer	0x40000	Maximum TRANSFER Size	True
pvid	00c1f1707355c8e50000000000000000	Physical volume identifier	False
queue_depth	3	Queue DEPTH	True
reserve_policy	no_reserve	Reserve Policy	True

Failover: MPIO on the client partition runs a fail_over algorithm. That means only one path is active at a time. If you shut down a Virtual I/O Server that serves the inactive path, then the path mode does not change to failed because no I/O is using this path.

Failover might require a small period of time in which the client re-establishes a path to the SAN. Testing of production workloads must be done in order to verify that this delay is acceptable.

4. Enable the virtual SCSI client adapter path timeout feature for each virtual SCSI client adapter. By default the feature is disabled (`vscsi_path_to=0`). To activate it use the **chdev** command as shown in the following example. In Example 16-56 the timeout is set to the minimum value of 30 seconds.

Example 16-56 Configuring the virtual SCSI client adapter path timeout feature

```
# chdev -l vscsi0 -a vscsi_path_to=30 -P
vscsi0 changed
# lsattr -El vscsi0
vscsi_err_recov delayed_fail N/A True
vscsi_path_to 30 Virtual SCSI Path Timeout True
```

Tip: If you try to set a value smaller than 30 seconds for the `vscsi_path_to` attribute, it will be automatically changed to 30 seconds.

5. Set the virtual SCSI adapter error recovery for each virtual SCSI client adapter to `fast_fail`. By default this feature is set `delayed_fail`. To activate it use the **chdev** command as shown in Example 16-57.

Example 16-57 Configuring the virtual client adapter error recovery feature

```
# chdev -l vscsi0 -a vscsi_err_recov=fast_fail -P
vscsi0 changed
# lsattr -El vscsi0
vscsi_err_recov fast_fail N/A True
vscsi_path_to 30 Virtual SCSI Path Timeout True
```

6. In our example we want the `DB_server` virtual SCSI traffic to go through `VIO_Server1` and the `Apps_server` traffic to through `VIO_Server2`. Determine which path is connected to which Virtual I/O Servers using the **lscfg** command and verify the slot numbers for the VSCSI devices, as shown in Example 16-58.

Example 16-58 Find out which parent belongs to which path

```
# lscfg -vI vscsi0
vscsi0          U9117.MMA.101F170-V3-C21-T1 Virtual SCSI Client Adapter

Hardware Location Code.....U9117.MMA.101F170-V3-C21-T1

# lscfg -vI vscsi1
vscsi1          U9117.MMA.101F170-V3-C22-T1 Virtual SCSI Client Adapter

Hardware Location Code.....U9117.MMA.101F170-V3-C22-T1
```

In our example slot number 21 points to VIO_Server1 and slot number 22 to VIO_Server2.

7. By default all the paths are defined with priority 1 meaning that traffic will go through the first path (Priority 1 is the highest priority, and you can define a priority from 1 to 255). For the DB_server partition no updates to the path priority are necessary because the first path is using vscsi0 which is connected to VIO_Server1. For the Apps_server partition the path priority has to be updated so that the primary path is going through VIO_Server2.

To set the primary path to use the VSCSI1 device, the priority of the VSCSI0 path is changed to 2 so that it becomes a lower priority, as shown in Example 16-59. The priority for the VSCSI1 devices remains at 0 so that it is the primary path.

Example 16-59 Changing the priority of a path

```
# chpath -l hdisk0 -p vscsi0 -a priority=2
path Changed
# lspath -AHE -l hdisk0 -p vscsi0
attribute value description user_settable

priority 2      Priority    True

# lspath -AHE -l hdisk0 -p vscsi1
attribute value description user_settable

priority 1      Priority    True
```

Tip: You might have to update the Preferred Path in the SAN to reflect the changed priority settings.

8. Reboot the client partition for the changes to take effect. Both changes require a reboot to be activated because this disk belongs to the rootvg and is in use.

Testing multipathing on AIX client

Perform the following steps to verify that the MPIO configuration works as expected. The steps here show how failover works when one Virtual I/O Server is shutdown. This can happen if you are performing maintenance on the Virtual I/O Server.

This example shows how path failover works in the DB_server partition.

1. Log in to the client partition and verify that all the paths are enabled using the **lspath** command as shown in Example 16-60.

Example 16-60 Displaying MPIO path status in the client partition

```
# lspath
Enabled hdisk0 vscsi0
Enabled hdisk0 vscsi1
```

2. Shut down VIO_Server2.
3. As soon as the failover to the alternate path has been done you will see a message in the errorlog, as shown in Example 16-61.

Example 16-61 Errorlog message when path fails

```
LABEL:          SC_DISK_ERR7
IDENTIFIER:      DE3B8540

Date/Time:       Sat Nov 24 15:21:39 CST 2007
Sequence Number: 10
Machine Id:      00C1F1704C00
Node Id:         db_server1
Class:           H
Type:            PERM
WPAR:            Global
Resource Name:   hdisk0
Resource Class:  disk
Resource Type:   vdisk
Location:        U9117.MMA.101F170-V3-C21-T1-L810000000000
```

Description
PATH HAS FAILED

Probable Causes
ADAPTER HARDWARE OR CABLE
DASD DEVICE

Failure Causes
UNDETERMINED

Recommended Actions
PERFORM PROBLEM DETERMINATION PROCEDURES
CHECK PATH

```

Detail Data
PATH ID
      1
SENSE DATA
0A00 2800 0017 0E38 0000 0804 0000 0000 0000 0000 0000 0000 0200
0B00 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000

```

The **lspath** command will show the path as failed, as shown in Example 16-62.

Example 16-62 lspath output with failed path

```

# lspath
Enabled hdisk0 vscsi0
Failed  hdisk0 vscsi1

```

4. Start the Virtual I/O Server.
5. When the Virtual I/O Server comes up the status of the failed path will go back to enabled if the `hcheck_interval` has been set. Use the **lspath** command as show in Example 16-63 to verify that all paths are enabled again.

Example 16-63 lspath output with enabled paths

```

# lspath
Enabled hdisk0 vscsi0
Enabled hdisk0 vscsi1

```

Tip: The path will only automatically go back to enabled status if the `hcheck_interval` has been defined. Else you have to manually set it back to enabled status using the **chpath** command as shown in the following example:

```
chpath -s enabled -l hdisk -p vscsi0
```

For this scenario, to show setting up IBM i client multipathing, we start from the basic configuration of the IBM i client configured with virtual SCSI LUNs from a single Virtual I/O Server as shown in 12.1, “Creating a Virtual I/O Server” on page 312.

LUNs: Though this example shows implementing IBM i multipathing for virtual SCSI LUNs, the same concept applies for virtual Fibre Channel LUNs using Virtual Fibre Channel with the considerations mentioned in “Multipathing in the IBM i client partition” on page 498.

Display Disk Configuration Status						
ASP Unit	Serial Number	Type	Model	Resource Name	Status	Hot Spare Protection
1					Unprotected	
	1 Y9UCTLXBVQ9G	6B22	050	DD001	Configured	N
	2 YW9FPXR5X759	6B22	050	DD004	Configured	N
	3 Y8VG3JUGRKLD	6B22	050	DD003	Configured	N
	4 YAP8GVNPCU7Z	6B22	050	DD002	Configured	N

Press Enter to continue.

F3=Exit F5=Refresh F9=Display disk unit details
F11=Disk configuration capacity F12=Cancel

Chapter 16. I/O virtualization setup **517**

Considerations:

- ▶ When dynamically adding a virtual SCSI adapter to the client partition or/and Virtual I/O Server partition using the Dynamic Logical Partitioning function, remember to change the partition profile as well to make the change persistent across partition restarts.
- ▶ For a dynamically added virtual Fibre Channel *client* adapter, remember to use the Save Current Configuration function to save to a new profile, and possibly make it the default profile, instead of changing the partition profile configuration to ensure that the generated virtual WWPNs are retained after the partition restarts.

In Figure 16-30, the virtual SCSI adapter is added to the partition profile.

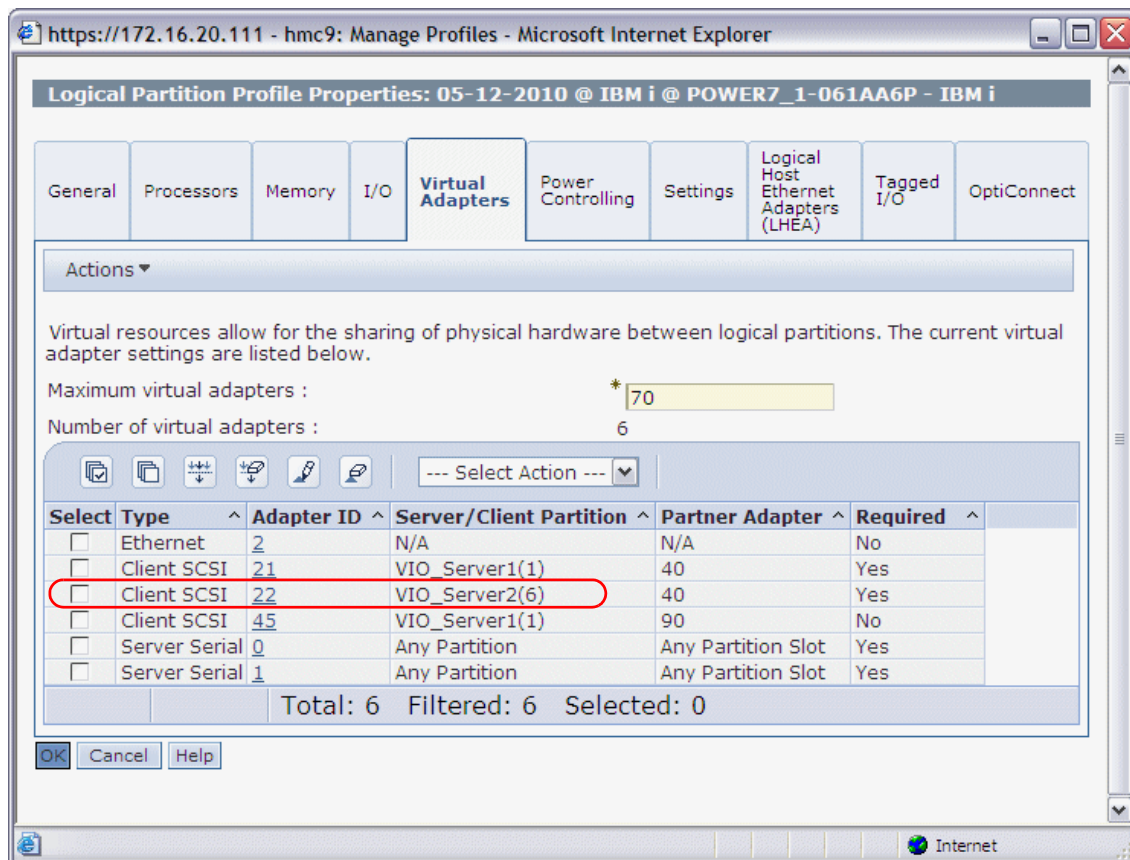


Figure 16-30 IBM i client partition with added virtual SCSI adapter for multipathing

After establishing the virtual SCSI adapter mapping for the second path, we verify again that the SAN storage volumes on both Virtual I/O Servers were changed to no_reserve policy as described in , “Configuring multipathing in the server” on page 503 and map them on the second Virtual I/O Server, as well to the IBM i client, using the **mkvdev** command as shown in Example 16-64.

Example 16-64 Volume mapping to the IBM i client on the second Virtual I/O Server

```
$ for i in 10 11 12 13 ; do lsdev -dev hdisk$i -attr reserve_policy ; done
value

no_reserve
value

no_reserve
value

no_reserve
value

no_reserve

$ lsmmap -vadapter vhost4
SVSA          Physloc                                Client Partition ID
-----
vhost4        U8233.E8B.061AA6P-V2-C40                0x000000006

VTD                                NO VIRTUAL TARGET DEVICE FOUND

$ mkvdev -vdev hdisk10 -vadapter vhost4 -dev vIBMi_LS
vIBMi_LS Available
$ mkvdev -vdev hdisk11 -vadapter vhost4 -dev vIBMi_2
vIBMi_2 Available
$ mkvdev -vdev hdisk12 -vadapter vhost4 -dev vIBMi_3
vIBMi_3 Available
$ mkvdev -vdev hdisk13 -vadapter vhost4 -dev vIBMi_4
vIBMi_4 Available
$ lsmmap -vadapter vhost4
```


SVSA	Physloc	Client Partition ID
vhost4	U8233.E8B.061AA6P-V2-C40	0x00000006
VTD	vIBMi_2	
Status	Available	
LUN	0x8200000000000000	
Backing device	hdisk11	
Physloc	U5802.001.0086848-P1-C3-T1-W500507630414C12C-L4010401900000000	
Mirrored	false	
VTD	vIBMi_3	
Status	Available	
LUN	0x8300000000000000	
Backing device	hdisk12	
Physloc	U5802.001.0086848-P1-C3-T1-W500507630414C12C-L4010401A00000000	
Mirrored	false	
VTD	vIBMi_4	
Status	Available	
LUN	0x8400000000000000	
Backing device	hdisk13	
Physloc	U5802.001.0086848-P1-C3-T1-W500507630414C12C-L4011401000000000	
Mirrored	false	
VTD	vIBMi_LS	
Status	Available	
LUN	0x8100000000000000	
Backing device	hdisk10	
Physloc	U5802.001.0086848-P1-C3-T1-W500507630414C12C-L4010401800000000	
Mirrored	false	

The IBM i client automatically detects the added second path for each of its virtual SCSI LUNs with one path seen from each Virtual I/O Server. Now displaying the IBM i disk configuration status from System Service Tools again, we can see that the resource name for the previously single-path disk units changed from DDxxx to DMPxxx, indicating the disk units are now multipath disk units as shown in Figure 16-31.

Display Disk Configuration Status						
ASP	Unit	Serial Number	Type	Model	Resource Name	Status
1						Unprotected
	1	Y9UCTLXBVQ9G	6B22	050	DMP003	Configured
	2	YW9FPXR5X759	6B22	050	DMP001	Configured
	3	Y8VG3JUGRKLD	6B22	050	DMP005	Configured
	4	YAP8GVNPCU7Z	6B22	050	DMP007	Configured
						Hot Spare Protection
						N
						N
						N
						N
						N
Press Enter to continue.						
F3=Exit F5=Refresh F9=Display disk unit details						
F11=Disk configuration capacity F12=Cancel						

Figure 16-31 IBM i SST Display disk configuration status

Using the IBM i System Service Tools option **3. Work with disk units** → **1. Display disk configuration** → **9. Display disk path status**, we can see that each disk unit enlists with two active paths as shown Figure 16-32.

Display Disk Path Status						
ASP	Unit	Serial Number	Type	Model	Resource Name	Path Status
1	1	Y9UCTLXBVQ9G	6B22	050	DMP003	Active
					DMP004	Active
1	2	YW9FPXR5X759	6B22	050	DMP001	Active
					DMP002	Active
1	3	Y8VG3JUGRKLD	6B22	050	DMP005	Active
					DMP006	Active
1	4	YAP8GVNPCU7Z	6B22	050	DMP007	Active
					DMP008	Active

Press Enter to continue.

F3=Exit F5=Refresh F9=Display disk unit details
F11=Display encryption status F12=Cancel

Figure 16-32 IBM i SST Display disk path status

Selecting the option **F9=Display disk unit details** from the Display Disk Path Status screen and looking at the Sys Card information shown for each path we can also verify that each disk unit is seen by a path from Virtual I/O Server 1, using virtual SCSI client adapter slot 21, and by a second path from Virtual I/O Server 2, using virtual SCSI client adapter slot 22, as shown in Figure 16-33.

Display Disk Unit Details

Type option, press Enter.

5=Display hardware resource information details

OPT	ASP	Unit	Serial Number	Sys Bus	Sys Card	Sys Board	I/O Adapter	I/O Bus	Ctl	Dev
1	1	Y9UCTLXBVQ9G		255	21	128		0	1	0
				255	22	128		0	1	0
1	2	YW9FPXR5X759		255	21	128		0	3	0
				255	22	128		0	3	0
1	3	Y8VG3JUGRKLD		255	21	128		0	2	0
				255	22	128		0	2	0
1	4	YAP8GVNPCU7Z		255	21	128		0	4	0
				255	22	128		0	4	0

F3=Exit

F9=Display disk units

F12=Cancel

Figure 16-33 IBM i SST Display disk unit details

This redundant Virtual I/O Server configuration using IBM i multipathing across two Virtual I/O Servers now protects the IBM i client from an outage of one Virtual I/O Server, for example, for Virtual I/O Server maintenance updates.

Testing multipathing on IBM i client

In the following test we verify the protection of the IBM i client partition, using multipathing across two Virtual I/O Servers, against Virtual I/O Server outages, by simulating a Virtual I/O Server outage with an immediate HMC power-down of the Virtual I/O Server partition.

After the simulated sudden outage of Virtual I/O Server 1 the IBM i client partition loses one path for each of its disk units reported by a CPPEA33 message **Warning - An external storage subsystem disk unit connection has failed.** as shown in Figure 16-34.

Additional Message Information

Message ID : CPPEA33 Severity : 70
Message type : Information
Date sent : 12/06/10 Time sent : 13:11:52

Message : **Warning - An external storage subsystem disk unit connection has failed.**
Cause : A connection from I/O adapter DC01 to external storage subsystem disk unit DMP007 has failed. There are still 1 active connections to this disk unit.
Recovery : Look for other errors related to this problem and report them to your hardware service provider.

Bottom

Press Enter to continue.

F3=Exit F6=Print F9=Display message details F12=Cancel
F21=Select assistance level

Figure 16-34 IBM i CPPEA33 message for a failed disk unit connection

Displaying the disk path status from IBM i System Service Tools again we can see that one path for the disk units, the one to Virtual I/O Server 1, is in failed status, while the path to Virtual I/O Server 2 remains active as shown in Figure 16-35.

Display Disk Path Status						
ASP	Unit	Serial Number	Type	Model	Resource Name	Path Status
1	1	Y9UCTLXBVQ9G	6B22	050	DMP003	Failed
					DMP004	Active
	2	YW9FPXR5X759			DMP001	Failed
					DMP002	Active
	3	Y8VG3JUGRKLD			DMP005	Failed
					DMP006	Active
	4	YAP8GVNPCU7Z			DMP007	Failed
					DMP008	Active
Press Enter to continue.						
F3=Exit F5=Refresh F9=Display disk unit details						
F11=Display encryption status F12=Cancel						

Figure 16-35 IBM i SST Display disk path status after outage of Virtual I/O Server1

Note that seeing a dump (SRC B600512D) for any non-load source virtual IOP in the IBM i Product Activity Log when a Virtual I/O Server partition goes away is an expected behavior and nothing to be concerned about.

After Virtual I/O Server 1 is operational again, the IBM i client almost instantly recognizes by its system-wide probes at 15 second intervals that the failed path by Virtual I/O Server 1 has become operational again and automatically starts using the path again as reported by message CPPEA35 Informational only. A connection to an external storage subsystem disk unit has been restored. as shown in Figure 16-36.

```
Additional Message Information

Message ID . . . . . : CPPEA35      Severity . . . . . : 40
Message type . . . . . : Information
Date sent . . . . . : 12/06/10      Time sent . . . . . : 13:27:52

Message . . . . . : Informational only. A connection to an external storage
subsystem disk unit has been restored.
Cause . . . . . : A connection from I/O adapter DC01 to external storage
subsystem disk unit DMP007 has been restored. There are now 2 active
connections to this disk unit.
Recovery . . . : No action required.

Bottom

Press Enter to continue.

F3=Exit  F6=Print  F9=Display message details  F12=Cancel
F21=Select assistance level
```

Figure 16-36 IBM i CPPEA35 message for a restored disk unit connection

Displaying the disk unit path status from IBM i System Service Tools now also shows both paths active again as shown in Figure 16-37.

Display Disk Path Status						
ASP	Unit	Serial Number	Type	Model	Resource Name	Path Status
1	1	Y9UCTLXBVQ9G	6B22	050	DMP003	Active
					DMP004	Active
1	2	YW9FPXR5X759	6B22	050	DMP001	Active
					DMP002	Active
1	3	Y8VG3JUGRKLD	6B22	050	DMP005	Active
					DMP006	Active
1	4	YAP8GVNPCU7Z	6B22	050	DMP007	Active
					DMP008	Active

Press Enter to continue.

F3=Exit

F5=Refresh

F9=Display disk unit details

F11=Display encryption status

F12=Cancel

Figure 16-37 IBM i SST Display disk path status

Linux client multipathing

The example in Figure 16-38 shows a configuration where the disks are located on an external storage subsystem in the SAN. The disks are assigned to both Virtual I/O Servers. The client partitions see the disks through two paths using device mapper multipath. Each of the paths is going through a different Virtual I/O Server.

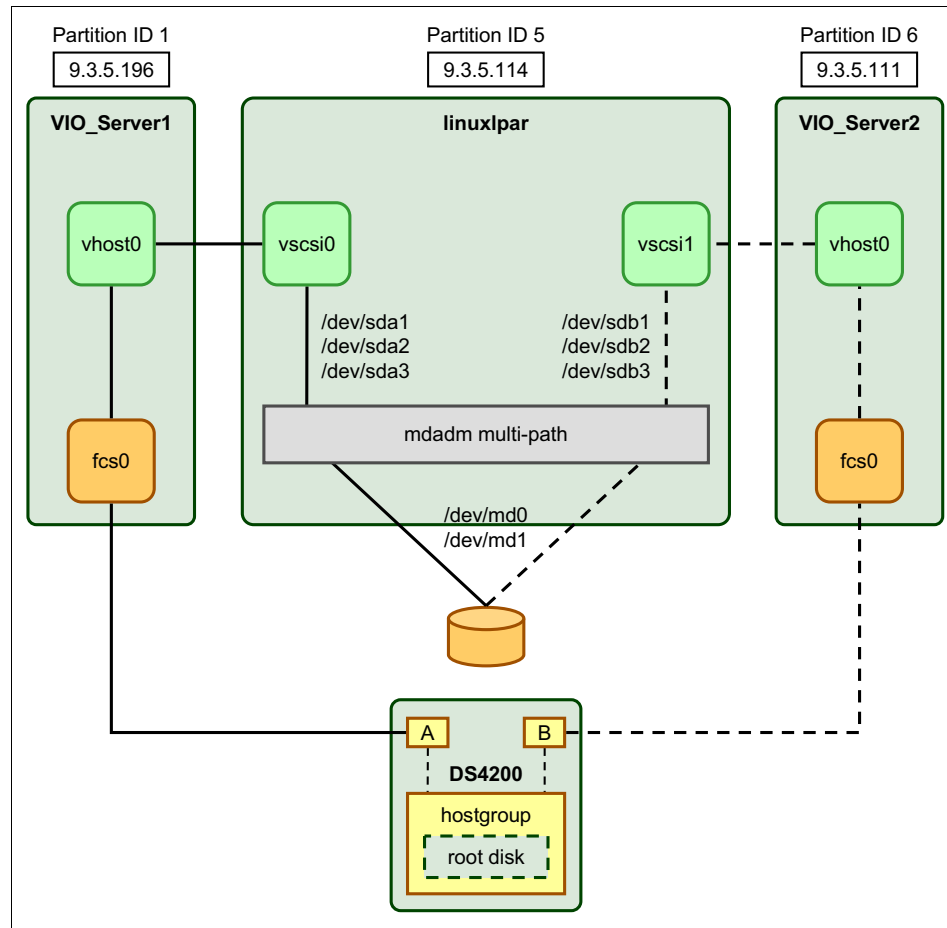


Figure 16-38 Linux client partition using MPIO to access SAN storage

To set up a Red Hat Linux 5 client partition with multipathing enabled, the **mpath** argument has to be specified when performing the initial Linux installation (see “Linux client partition installation” on page 378).

To install SuSE Linux Enterprise Server 10 SP3 and prior releases on a multipath device, */boot* must be created as a non-multipathed partition. More details on that topic can be found at the following website:

http://www.novell.com/documentation/sles10/stor_admin/?page=/documentation/sles10/stor_admin/data/mpiotools.html

Tip: On Red Hat Enterprise Linux 6, release there is no need to specify the **mpath** argument on the installation because multipath is enabled by default at the boot time. SuSE Linux Enterprise Server 11 has full multipath support as well.

The main advantage of the MPIO configuration compared to RAID mirroring is that after the failure, the lost paths will be automatically recovered when they are available again. There is no need to log in to a client partition and take corrective actions.

Another advantage of using multipath instead of Linux Software RAID mirroring is that the multipath solution requires half of the storage space compared to the mirroring solution.

Use the **multipathd -k** command to start the interactive mode. The multipath configuration can also be changed directly at */etc/multipath.conf* file. Note that the **multipathd** daemon must be restarted to make the changes effective. The following example shows a simple configuration with *sd*c device as a non-multipathed disk.

```
blacklist {
    devnode "sd*"
}

defaults {
    path_checker readsector0
    user_friendly_names yes
}
```

Tip: The option *user_friendly_names* must be set in order to have short names for the paths such as *mpath0*.

The device mapper multipath devices can be listed using the **multipath -ll** command. For a more verbose output, use **multipath -v3**.

Testing multipathing on the Linux client

To verify that your device mapper multipath configuration works as expected, perform the following steps:

1. Verify that both paths are active using the **multipath** command:

```
# multipath -ll
mpath0 (3600a0b80000bdc160000052847436d1e) dm-0 AIX,VDasD
[size=20G][features=0][hwhandler=0]
\_ round-robin 0 [prio=1][active]
  \_ 0:0:1:0 sda 8:0 [active][ready]
  \_ round-robin 0 [prio=1][enabled]
    \_ 1:0:1:0 sdb 8:16 [active][ready]
```

Tip: For path management, you can also use the interactive shell, which can be started using the **multipathd -k** command. You can enter an interactive command, you can enter **help** to get a list of available commands, or you can enter CTRL-D to quit.

2. Shut down one of the Virtual I/O Servers. In this example VIO_Server1 is shut down. In `var/log/messages` you will see the following messages:

```
ibmvscsi: Partner adapter not ready
ibmvscsi: error after reset
Dec 7 15:46:57 localhost kernel: ibmvscsi: Virtual adapter failed rc 2!
Dec 7 15:46:57 localhost kernel: rpa_vscsi: SPR_VERSION: 16.a
Dec 7 15:46:57 localhost kernel: ibmvscsi: Partner adapter not ready
Dec 7 15:46:57 localhost kernel: ibmvscsi: error after reset
device-mapper: multipath: Failing path 8:0.
Dec 7 15:47:01 localhost kernel: device-mapper: multipath: Failing path 8:0.
Dec 7 15:47:01 localhost multipathd: 8:0: reinstated
Dec 7 15:47:01 localhost multipathd: mpath0: remaining active paths: 2
Dec 7 15:47:01 localhost multipathd: sda: readsector0 checker reports path is down
Dec 7 15:47:01 localhost multipathd: checker failed path 8:0 in map mpath0
Dec 7 15:47:01 localhost multipathd: mpath0: remaining active paths: 1
Dec 7 15:47:06 localhost multipathd: sda: readsector0 checker reports path is down
```

3. When using the **multipath** command to display the path status, one path must be marked as failed as shown here:

```
# multipath -ll
sda: checker msg is "readsector0 checker reports path is down"
mpath0 (3600a0b80000bdc160000052847436d1e) dm-0 AIX,VDASD
[size=20G][features=0][hwandler=0]
\_ round-robin 0 [prio=0][enabled]
  \_ 0:0:1:0 sda 8:0   [failed][faulty]
\_ round-robin 0 [prio=1][active]
  \_ 1:0:1:0 sdb 8:16  [active][ready]
[root@localhost ~]#
```

4. Restart the Virtual I/O Server. After the Virtual I/O Server is rebooted, you will see the following messages in `/var/adm/messages`:

```
ibmvscsic: sent SRP login
ibmvscsi: host srp version: 16.a, host partition VIO_Server1 (1), OS 3, max io
1048576
Dec  7 15:50:36 localhost last message repeated 9 times
Dec  7 15:50:39 localhost kernel: ibmvscsi: partner initialized
Dec  7 15:50:39 localhost kernel: ibmvscsic: sent SRP login
Dec  7 15:50:39 localhost kernel: ibmvscsi: SRP_LOGIN succeeded
Dec  7 15:50:39 localhost kernel: ibmvscsi: host srp version: 16.a, host partition
VIO_Server1 (1), OS 3, max io 1048576
Dec  7 15:50:41 localhost multipathd: sda: readsector0 checker reports path is down
Dec  7 15:50:46 localhost multipathd: sda: readsector0 checker reports path is up
Dec  7 15:50:46 localhost multipathd: 8:0: reinstated
Dec  7 15:50:46 localhost multipathd: mpath0: remaining active paths: 2
```

5. When using the **multipath** command to display the path status, both paths must be active again:

```
# multipath -ll
mpath0 (3600a0b80000bdc160000052847436d1e) dm-0 AIX,VDASD
[size=20G][features=0][hwandler=0]
\_ round-robin 0 [prio=1][enabled]
  \_ 0:0:1:0 sda 8:0   [active][ready]
\_ round-robin 0 [prio=1][active]
  \_ 1:0:1:0 sdb 8:16  [active][ready]
```

Using Virtual Fibre Channel for multipathing

The multipath configuration using Virtual Fibre Channel is basically the same as for a physical Fibre Channel device. To set up a multipath device using Virtual Fibre Channel you need to assign the virtual Fibre Channels' WWPNs to the SAN logical drives. Virtual Fibre Channel's WWPNs can be retrieved from the partition profile at the HMC interface. Additionally, you can list the WWPNs directly from Linux as follows:

```
root@Power7-2-RHEL ~]# cat /sys/class/fc_host/host5/port_name
0xc0507603039e0003
```

Two WWPNs for each Virtual Fibre Channel port are required in order to support Live Partition Mobility. When configuring the SAN, you must be sure to assign both WWPNs to each LUN. In the case of multipath, that will be a total of 4 WWPNs as follows:

```
0xc0507603039e0002
0xc0507603039e0003
0xc0507603039e0008
0xc0507603039e0009
```

After the mappings are done on the Virtual I/O Server and the WWPNs are assigned to the SAN disk, the paths are automatically detected by Linux.

```
[root@Power7-2-RHEL ~]# lsscsi -v|grep rport
5:0:0:0]    disk    IBM      2107900          .278  /dev/sdc
    dir: /sys/bus/scsi/devices/5:0:0:0
[/sys/devices/vio/30000038/host5/rport-5:0-0/target5:0:0/5:0:0:0]
[6:0:0:0]    disk    IBM      2107900          .278  /dev/sdd
    dir: /sys/bus/scsi/devices/6:0:0:0
[/sys/devices/vio/30000039/host6/rport-6:0-0/target6:0:0/6:0:0:0]
```

The multipath devices can be listed with **multipath -ll** command as follows:

```
[root@Power7-2-RHEL ~]# multipath -ll
mpath1 (36005076304ffc12c0000000000001020) dm-2 IBM,2107900
[size=15G][features=0][hwhandler=0][rw]
\_ round-robin 0 [prio=2][active]
  \_ 5:0:0:0 sdc 8:32 [active][ready]
  \_ 6:0:0:0 sdd 8:48 [active][ready]
```

Tip: The `fast_io_fail_tmo` attribute of the `fc_remote_port` can be enabled in order to change the time it takes to failover when a path fails. The setup differs from distribution to another and it can be enabled by default. To change the timeout to 5 seconds set `fast_io_fail_tmo` as follows:

```
echo 5 > /sys/class/fc_remote_ports/rport-3\:0-0/fast_io_fail_tmo
```

16.2.7 Availability configurations using mirroring

This scenario shows how to set up a highly available virtual SCSI or virtual Fibre Channel (VFC) configuration using mirroring in the client partition. In our example shown in Figure 16-39 each client partition is configured with two virtual SCSI adapters. Each of these virtual adapters is connected to a different Virtual I/O Server and provides virtual SCSI LUNs to the client partition.

On the Virtual I/O Servers, the virtual SCSI disks for the AIX, and Linux client partitions are backed by a logical volume while for the IBM i client partition the virtual LUNs are backed by LUNs from two different SAN storage system. The AIX, and Linux client partitions use LVM mirroring so that the rootvg disk access is redundant. The IBM i client partition uses IBM i mirroring of its SYSBAS disk space for protection against both, Virtual I/O Server and SAN storage system outages.

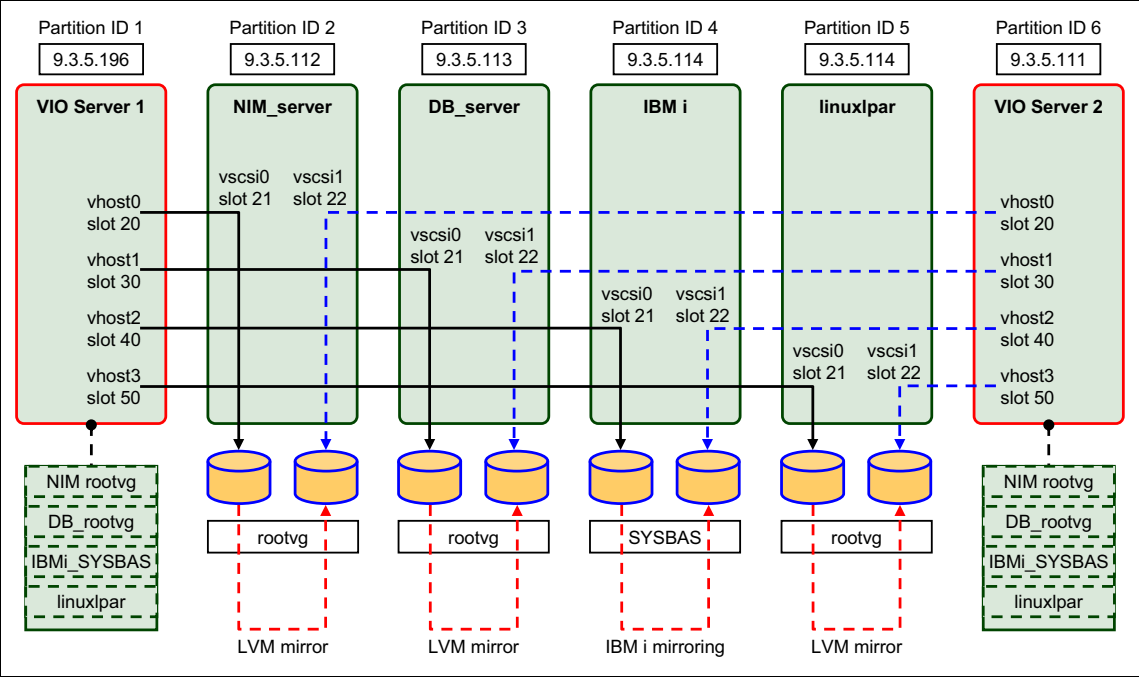


Figure 16-39 Redundant Virtual I/O Server client mirroring scenario

Configuring the Virtual I/O Server for client mirroring

Use the following steps to set up the scenario:

1. Create two Virtual I/O Server partitions and name them VIO_Server1 and VIO_Server2, following the instructions in 12.1, “Creating a Virtual I/O Server” on page 312. In step 10, select one Ethernet Adapter and one storage adapter. Depending on the hardware configuration you have this might be a Fibre Channel adapter or a SCSI adapter.

Figure 16-40 shows the adapters that were selected in the example configuration for VIO_Server2. Slot C5 on bus 514 contains an Ethernet adapter and slot c3 on bus 518 contains a Fibre Channel adapter.

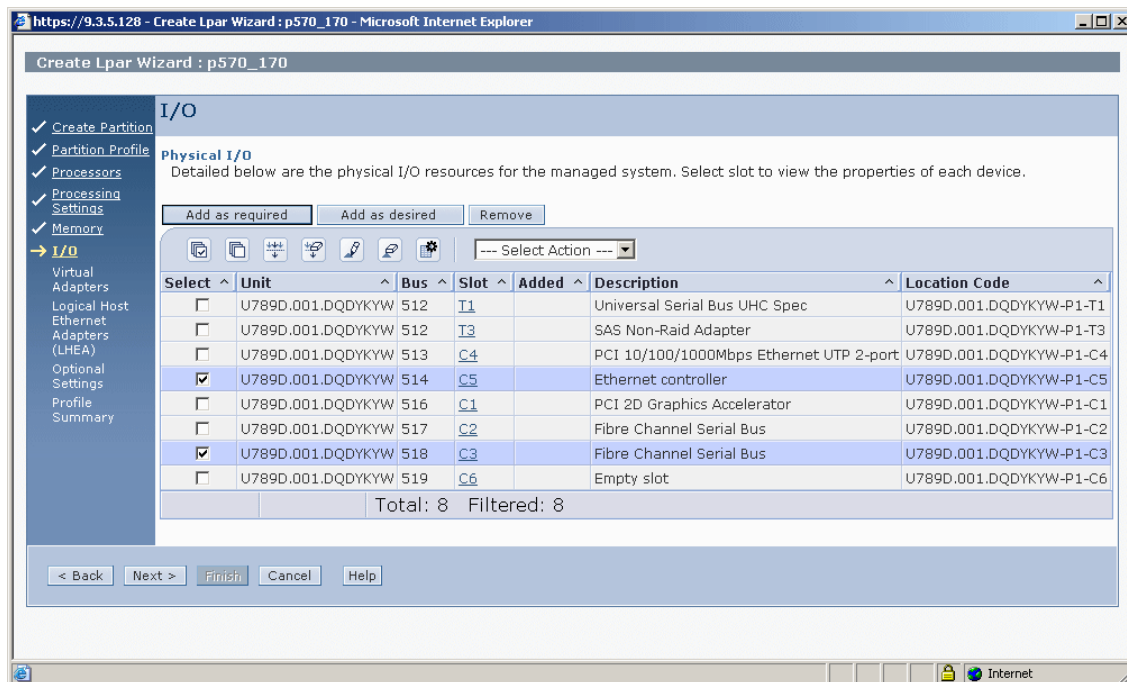


Figure 16-40 VIO_Server2 physical adapter selection

2. Install both Virtual I/O Servers by following the instructions in 12.2, “Installation of Virtual I/O Server” on page 333.
3. Configure the virtual SCSI adapters on VIO_Server1 and VIO_Server2 as shown in Table 16-7.

Table 16-7 Virtual SCSI adapter configuration for LVM mirroring

Virtual I/O Server	Virtual I/O Server slot	Client partition	Client partition slot
VIO_Server1	20	NIM_server	21
VIO_Server1	30	DB_server	21
VIO_Server1	40	IBM i	21
VIO_Server1	50	linuxlpar	21
VIO_Server2	20	NIM_server	22
VIO_Server2	30	DB_server	22
VIO_Server2	40	IBM i	22
VIO_Server2	50	linuxlpar	22

4. Create the client partitions as shown in Figure 16-39 on page 534 following the instructions in “Creating a Virtual I/O Server” on page 312. Each client partition needs to be configured with two virtual SCSI adapters as shown in Table 16-7 on page 536. Figure 16-41 shows the virtual SCSI adapter configuration for VIO_Server2 on the HMC.

Tip: In Figure 16-41 there is a filter applied to the *Type* column so that only the virtual SCSI adapters are shown.

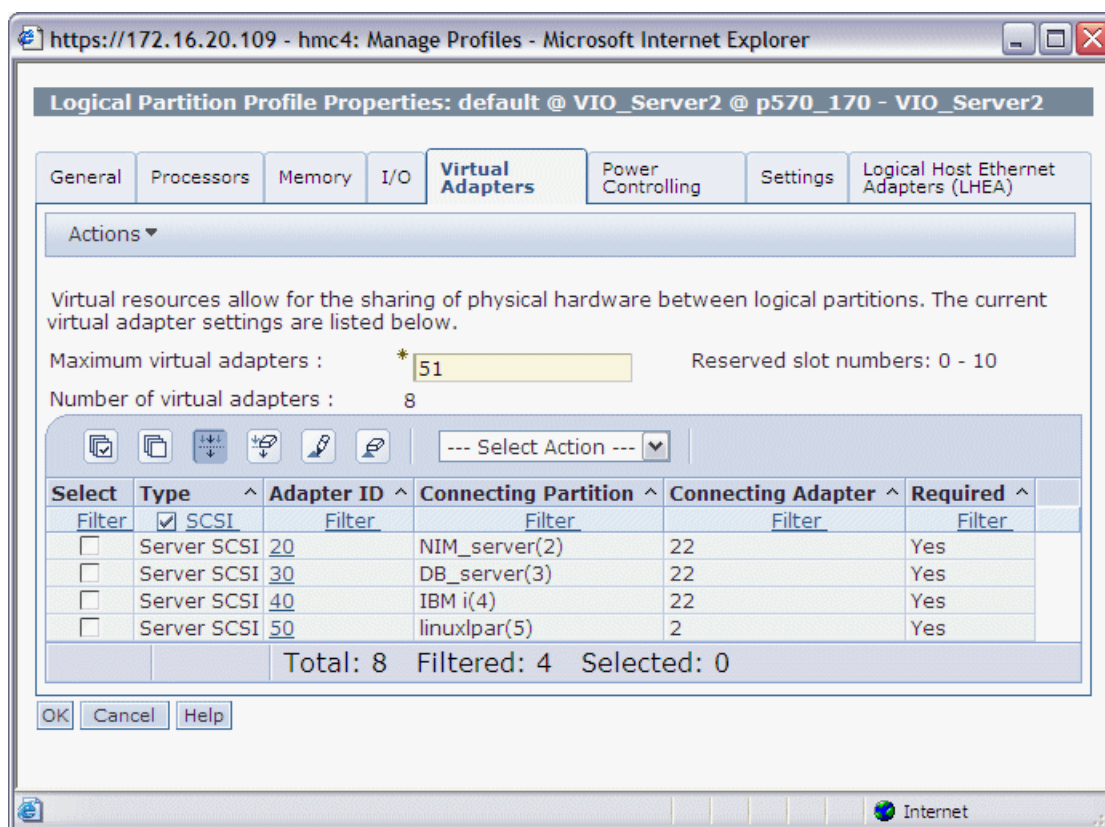


Figure 16-41 Virtual SCSI adapters for VIO_Server2

5. Also add one or two virtual Ethernet adapters to each client to provide the highly available network access: One adapter if you plan on using SEA failover for network redundancy, as described in “SEA failover” on page 591, or two adapters if you plan on using Network Interface Backup instead, as described in 16.3.3, “EtherChannel Backup in the AIX client” on page 603.

6. Create the volume group and logical volumes on VIO_Server1 and VIO_Server2 for the AIX, and Linux client partitions. See “Defining virtual disks” on page 466 for more information.
7. Check the devices list on VIO_Server1 and VIO_Server2 using the **lsdev** command, as in Example 16-65. There must be four vhost devices available, that is, one virtual SCSI server adapter for connection of each client partition.

Example 16-65 VIO_Server2 partition

```
$ lsdev -virtual
name          status
description
ent1           Available Virtual I/O Ethernet Adapter (1-lan)
ent2           Available Virtual I/O Ethernet Adapter (1-lan)
vasi0          Available Virtual Asynchronous Services Interface (VASI)
vhost0        Available Virtual SCSI Server Adapter
vhost1        Available Virtual SCSI Server Adapter
vhost2        Available Virtual SCSI Server Adapter
vhost3        Available Virtual SCSI Server Adapter
vsa0           Available LPAR Virtual Serial Adapter
vappssrv_rvg   Available Virtual Target Device - Logical Volume
vdbsrv_rvg     Available Virtual Target Device - Logical Volume
vlinux         Available Virtual Target Device - Logical Volume
vnimsrv_rvg    Available Virtual Target Device - Logical Volume
```

8. On each of two Virtual I/O Servers, an equal size logical volume from the rootvg_clients volume group is mapped to each of the AIX and Linux client partitions, and four equal size SAN storage LUNs are mapped to the IBM i client partition, as shown in Example 16-66.

Example 16-66 Virtual SCSI mappings

```
$ mkvdev -vdev nimsrv_rvg -vadapter vhost0 -dev vnimsrv_rvg
vnimsrv_rvg Available
$ mkvdev -vdev dbsrv_rvg -vadapter vhost1 -dev vdbsrv_rvg
vdbsrv_rvg Available
$ mkvdev -vdev hdisk23 -vadapter vhost2 -dev vIBMi_LS
vIBMi_LS Available
$ mkvdev -vdev hdisk24 -vadapter vhost2 -dev vIBMi_2
vIBMi_2 Available
$ mkvdev -vdev hdisk25 -vadapter vhost2 -dev vIBMi_3
vIBMi_3 Available
$ mkvdev -vdev hdisk26 -vadapter vhost2 -dev vIBMi_4
vIBMi_4 Available
$ mkvdev -vdev linux -vadapter vhost3 -dev vlinux
vlinux Available
$ lsmmap -all
```

SVSA	Physloc	Client Partition ID

vhost0	U9117.MMA.101F170-V6-C20	0x00000000
VTD	vnimsrv_rvg	
Status	Available	
LUN	0x8100000000000000	
Backing device	nimsrv_rvg	
Physloc		
SVSA	Physloc	Client Partition
ID		

vhost1	U9117.MMA.101F170-V6-C30	0x00000000
VTD	vdbsrv_rvg	
Status	Available	
LUN	0x8100000000000000	
Backing device	dbsrv_rvg	
Physloc		
SVSA	Physloc	Client Partition
ID		

vhost2	U9117.MMA.101F170-V6-C40	0x00000000
VTD	vIBMi_2	
Status	Available	
LUN	0x8200000000000000	
Backing device	hdisk11	
Physloc		
U5802.001.0086848-P1-C2-T1-W500507630410412C-L4010401900000000		
Mirrored	false	
VTD	vIBMi_3	
Status	Available	
LUN	0x8300000000000000	
Backing device	hdisk12	
Physloc		
U5802.001.0086848-P1-C2-T1-W500507630410412C-L4010401A00000000		
Mirrored	false	
VTD	vIBMi_4	
Status	Available	
LUN	0x8400000000000000	
Backing device	hdisk13	
Physloc		
U5802.001.0086848-P1-C2-T1-W500507630410412C-L4011401000000000		
Mirrored	false	

VTD	vIBMi_LS	
Status	Available	
LUN	0x8100000000000000	
Backing device	hdisk10	
Physloc	U5802.001.0086848-P1-C2-T1-W500507630410412C-L4010401800000000	
Mirrored	false	
SVSA ID	Physloc	Client Partition

vhost3	U9117.MMA.101F170-V6-C50	0x00000000
VTD	vlinux	
Status	Available	
LUN	0x8100000000000000	
Backing device	linux	
Physloc		

AIX client LVM mirroring

In this section we provide mirroring scenarios and discuss testing procedures.

Mirroring scenarios

After completing the Virtual I/O Server configuration for client mirroring, when you bring up the AIX client partition, you must have hdisk0 and hdisk1 available, as shown in Example 16-67. Mirror the rootvg as you normally do on AIX.

Example 16-67 NIM_server partition with hdisk1 served off of VIO_Server2

```
# hostname
NIM_server
# lspv
hdisk0          00c1f1706e8787f9          rootvg          active
hdisk1          none                    None
# lsdev -Cc disk
hdisk0 Available Virtual SCSI Disk Drive
hdisk1 Available Virtual SCSI Disk Drive
# lsdev -p vscsi0
hdisk0 Available Virtual SCSI Disk Drive
# lsdev -p vscsi1
hdisk1 Available Virtual SCSI Disk Drive
# extendvg rootvg hdisk1
0516-1254 extendvg: Changing the PVID in the ODM.
# mirrorvg -m rootvg hdisk1
0516-1804 chvg: The quorum change takes effect immediately.
```

```

0516-1126 mirrorvg: rootvg successfully mirrored, user should perform
        bosboot of system to initialize boot records. Then, user must modify
        bootlist to include: hdisk0 hdisk1.
# bosboot -a -d /dev/hdisk1

bosboot: Boot image is 26927 512 byte blocks.
# bootlist -m normal hdisk0 hdisk1
# bootlist -m normal -o
hdisk0 blv=hd5
hdisk1 blv=hd5

```

Testing LVM mirroring on the AIX client

Perform the following steps to test if the LVM mirroring configuration works as expected:

1. Make sure that all logical volumes in the rootvg volume group are correctly mirrored. One mirror copy must be on the virtual disk provided by VIO_Server1, while the other needs to be on the virtual disk provided by VIO_Server2.
2. Verify that all the logical volumes are synchronized and that you have no stale partitions, as shown in Example 16-68.

Example 16-68 Verifying synchronization status using the lsvg command

```

# lsvg -l rootvg
rootvg:
LV NAME          TYPE      LPs      PPs      PVs  LV STATE    MOUNT POINT
hd5              boot      3         6         2  closed/syncd  N/A
hd6              paging    64        128        2  open/syncd    N/A
hd8              jfs2log   1          2          2  open/syncd    N/A
hd4              jfs2      3          6          2  open/syncd    /
hd2              jfs2     82        164          2  open/syncd    /usr
hd9var           jfs2      2          4          2  open/syncd    /var
hd3              jfs2      8         16          2  open/syncd    /tmp
hd1              jfs2      2          4          2  open/syncd    /home
hd10opt          jfs2     10         20          2  open/syncd    /opt
hd11admin        jfs2     16         32          2  open/syncd    /admin

```

Important: If your Virtual I/O Servers also provide Shared Ethernet Adapters, then Shared Ethernet Adapter of Network Interface Backup failovers will occur on the VIO clients when a Virtual I/O Server shuts down or fails.

3. Shut down the VIO_Server2 partition using the **shutdown** command.
4. When VIO_Server2 is down, the hdisks that are served by this Virtual I/O Server must go to missing state in the client partitions. You will also start to see stale partitions as shown in Example 16-69 in the errorlog. The client partitions must continue to be available and not experience a crash.

Example 16-69 Errors when virtual disk becomes unavailable

```
# errpt
IDENTIFIER  TIMESTAMP  T C RESOURCE_NAME  DESCRIPTION
EAA3D429   1123163907 U S LVDD          PHYSICAL PARTITION MARKED STALE
F7DDA124   1123163907 U H LVDD          PHYSICAL VOLUME DECLARED MISSING
52715FA5   1123163907 U H LVDD          FAILED TO WRITE VOLUME GROUP STATUS
AREA
E86653C3   1123163907 P H LVDD          I/O ERROR DETECTED BY LVM
EAA3D429   1123163907 U S LVDD          PHYSICAL PARTITION MARKED STALE
E86653C3   1123163907 P H LVDD          I/O ERROR DETECTED BY LVM
857033C6   1123163907 T S vscsil        Underlying transport error
# lspv hdisk1
PHYSICAL VOLUME:      hdisk1              VOLUME GROUP:      rootvg
PV IDENTIFIER:        00c1f1706e9e4ca9 VG IDENTIFIER
00c1f17000004c00000001166e8788c9
PV STATE:             missing
STALE PARTITIONS:     16                  ALLOCATABLE:       yes
PP SIZE:              8 megabyte(s)        LOGICAL VOLUMES:   10
TOTAL PPs:            639 (5112 megabytes)   VG DESCRIPTORS:    1
FREE PPs:             448 (3584 megabytes)   HOT SPARE:         no
USED PPs:             191 (1528 megabytes)   MAX REQUEST:       256 kilobytes
FREE DISTRIBUTION:     125..64..03..128..128
USED DISTRIBUTION:     03..64..124..00..00
#
```

5. Reactivate the VIO_Server2 partition from the HMC. When it is up, issue a **varyonvg rootvg** command in the client partition. This will put the missing hdisk back into active state and synchronize the stale partitions as shown in Example 16-70. Depending on the size of the volume group and the number of stale partitions, it might take a few minutes until all logical volumes are synchronized. You can also issue the **varyonvg -n rootvg** command and then start the synchronization manually with the **syncvg -v rootvg** command.

Important: If you have several volume groups in a client partition that are served by Virtual I/O Server, make sure that you issue a **varyonvg** for each of them and verify each one if there are no stale partitions.

Tip: Depending on your dump device configuration, you might receive the following error message when issuing the **varyonvg rootvg** command:

0516-1774 varyonvg: Cannot varyon volume group with an active dump device on a missing physical volume.
Use sysdumpdev to temporarily replace the dump device with /dev/sysdumpnull and try again.

In this case, set the dump device temporarily to /dev/sysdumpnull and restore the original dump device configuration after the varyonvg.

Example 16-70 shows how to add the missing disk back.

Example 16-70 Adding the missing disk back

```
# varyonvg rootvg
# lspv hdisk1
PHYSICAL VOLUME:    hdisk1                VOLUME GROUP:    rootvg
PV IDENTIFIER:      00c1f1706e9e4ca9 VG IDENTIFIER
00c1f17000004c00000001166e8788c9
PV STATE:           active
STALE PARTITIONS:    0                     ALLOCATABLE:      yes
PP SIZE:             8 megabyte(s)         LOGICAL VOLUMES:  10
TOTAL PPs:           639 (5112 megabytes)   VG DESCRIPTORS:   1
FREE PPs:            448 (3584 megabytes)    HOT SPARE:        no
USED PPs:            191 (1528 megabytes)    MAX REQUEST:      256 kilobytes
FREE DISTRIBUTION:   125..64..03..128..128
USED DISTRIBUTION:   03..64..124..00..00
# lsvg -l rootvg
rootvg:
LV NAME            TYPE      LPs    PPs    PVs  LV STATE    MOUNT POINT
hd5                 boot      3      6      2    closed/syncd N/A
hd6                 paging    64     128    2    open/syncd   N/A
hd8                 jfs2log   1      2      2    open/syncd   N/A
hd4                 jfs2      3      6      2    open/syncd   /
hd2                 jfs2     82     164    2    open/syncd   /usr
hd9var              jfs2      2      4      2    open/syncd   /var
hd3                 jfs2      8      16     2    open/syncd   /tmp
hd1                 jfs2      2      4      2    open/syncd   /home
hd10opt             jfs2     10     20     2    open/syncd   /opt
hd11admin           jfs2     16     32     2    open/syncd   /admin
```

IBM i client mirroring

In this section we provide mirroring scenarios and discuss testing procedures.

Mirroring scenarios

This scenario describes how to implement mirroring on the IBM i client across two Virtual I/O Servers as shown in Figure 16-39 on page 534.

- Before starting IBM i mirroring, we display the current IBM i disk configuration, which consists of a single virtual SCSI LUN as the load source provided by a single Virtual I/O Server only, by running the IBM i CL command STRSST and logging in to IBM i System Service Tools (SST). We select the option, **3. Work with disk units** → **1. Display disk configuration** → **1. Display disk configuration status** as shown in Figure 16-42.

Display Disk Configuration Status							
Serial		Resource				Hot Spare	
ASP Unit	Number	Type	Model	Name	Status		
Protection							
1					Unprotected		
1	Y9UCTLXBVQ9G	6B22	050	DD001	Configured	N	
Press Enter to continue.							
F3=Exit		F5=Refresh		F9=Display disk unit details			
F11=Disk configuration capacity				F12=Cancel			

Figure 16-42 IBM i SST Display disk configuration status

2. After mapping the virtual LUNs to the IBM i client partition as described in , “Configuring the Virtual I/O Server for client mirroring” on page 535, the newly mapped virtual LUNs from Virtual I/O Server 1 and Virtual I/O Server 2 show up as *non-configured* units on the IBM i client, which can be displayed by selecting the option **4. Display non-configured units** from the Display Disk Configuration SST screen as shown in Figure 16-43.

Display Non-Configured Units					
Serial	Resource				
Number	Type	Model	Name	Capacity	Status
YDU8UT78ZHMZ	6B22	050	DPH003	19088	Non-configured
Y2LVHS2WFVCM	6B22	050	DPH001	19088	Non-configured
Y6FWEN7UP9DW	6B22	050	DPH002	19088	Non-configured
YY8TMA75JZTR	6B22	050	DPH004	19088	Non-configured
YZG9ZK2YKVV4	6B22	050	DPH005	19088	Non-configured
YYMD6NS9YGL4	6B22	050	DPH006	19088	Non-configured
Y5UQXAAMRRYR	6B22	050	DPH007	19088	Non-configured
Press Enter to continue.					
F3=Exit F5=Refresh F9=Display disk unit details					
F11=Display device parity status F12=Cancel					

Figure 16-43 IBM i SST Display non-configured units

3. To figure out for each non-configured disk unit from which Virtual I/O Server it is provided, we press **F9=Display disk unit details** to look at the Sys Card information as shown in Figure 16-44, which corresponds to the virtual SCSI client adapter slot and shows us the three remaining disk units provided by Virtual I/O Server 1 (Sys Card = 21) and the four disk units provided by Virtual I/O Server 2 (Sys Card = 22).

Important: Currently all virtual SCSI or Fibre Channel adapters report in on IBM i under the same bus number 255, which allows for *IOP-level* mirrored protection only. To implement the concept of *bus-level* mirrored protection for virtual LUNs with larger configurations having more than one virtual IOP per mirror side, in order not to compromise redundancy, consider iteratively adding LUNs from one IOP pair at a time to the auxiliary storage pool by selecting the LUNs from one virtual IOP from each mirror side.

Display Disk Unit Details									
Type option, press Enter.									
5=Display hardware resource information details									
OPT	ASP	Unit	Serial Number	Sys Bus	Sys Card	Sys Board	I/O Adapter	I/O Bus	Dev
	*	*	YDU8UT78ZHMZ	255	22	128		0	3
	*	*	Y2LVHS2WFVCM	255	22	128		0	1
	*	*	Y6FWEN7UP9DW	255	22	128		0	2
	*	*	YY8TMA75JZTR	255	22	128		0	4
	*	*	YZG9ZK2YKVV4	255	21	128		0	2
	*	*	YYMD6NS9YGL4	255	21	128		0	3
	*	*	Y5UQXAAMRRYR	255	21	128		0	4
F3=Exit F9=Display disk units F12=Cancel									

Figure 16-44 IBM i SST Display disk unit details

4. Before we are able to start mirrored protection for the IBM i system auxiliary storage pool (ASP1), we need to have an even number of disk units configured in the ASP because each mirrored disk unit requires two sub-units A and B for each mirror side. Pressing **F12=Cancel** twice to get back to the Work with Disk Units screen, we select **2. Work with Disk Configuration** → **2. Add units to ASPs** → **3. Add units to existing ASPs**, then select the first virtual SCSI LUN (Ctl = 1) from Virtual I/O Server 2 (Sys Card = 22), that is, the second one displayed in the list (see Figure 16-44 on page 546), to be added to ASP 1 as shown in Figure 16-45 and press Enter.

Tip: We add a single disk unit at this time only, to minimize the time for synchronization at the subsequently required IPL to activate mirrored protection. Any further disk units can then be added to the mirrored ASP concurrently from SST when the IPL completed and IBM i is available again.

Specify ASPs to Add Units to						
Specify the existing ASP to add each unit to.						
Specify	Serial	Type	Model	Capacity	Resource	
ASP	Number				Name	
1	YDU8UT78ZHMZ	6B22	050	19088	DPH003	
	Y2LVHS2WFVCM	6B22	050	19088	DPH001	
	Y6FWEN7UP9DW	6B22	050	19088	DPH002	
	YY8TMA75JZTR	6B22	050	19088	DPH004	
	YZG9ZK2YKVV4	6B22	050	19088	DPH005	
	YYMD6NS9YGL4	6B22	050	19088	DPH006	
	Y5UQXAAMRRYR	6B22	050	19088	DPH007	

F3=Exit F5=Refresh F11=Display disk configuration capacity
F12=Cancel

Figure 16-45 IBM i SST Specify ASPs to add units to

5. We confirm the warning message that the disk unit might possibly be configured for another operating system by pressing **F10=Ignore problems and continue** as shown in Figure 16-46.

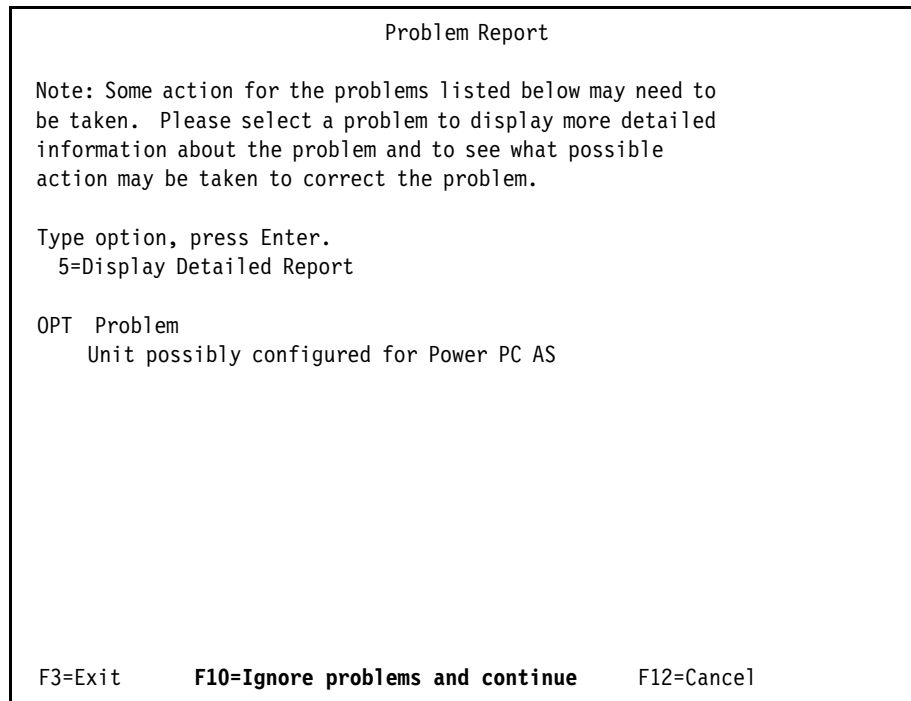


Figure 16-46 IBM i SST Problem Report Unit possibly configured for Power PC AS

6. The resulting ASP configuration after completing the Add to ASP function is shown for verification which we confirm by pressing Enter to continue as shown in Figure 16-47.

Confirm Add Units

Add will take several minutes for each unit. The system will have the displayed protection after the unit(s) are added.

Press Enter to confirm your choice for Add units.

Press F9=Capacity Information to display the resulting capacity.

Press F10=Confirm Add and Balance data on units.

Press F12=Cancel to return and change your choice.

ASP Unit	Serial Number	Type	Model	Resource Name	Protection	Hot Spare Protection
1					Unprotected	
	1 Y9UCTLXBVQ9G	6B22	050	DD001	Unprotected	N
	2 Y2LVHS2WFVCM	6B22	050	DPH001	Unprotected	N

F9=Resulting Capacity

F10=Add and Balance

F11=Display Encryption Status

F12=Cancel

Figure 16-47 IBM i SST Confirm Add Units

While being added to the ASP, the percentage progress for the new disk units being initialized, that is, formatted, and configured for usage by IBM i is displayed until its completion message as shown in Figure 16-48.

Add Units to ASPs

Select one of the following:

1. Create unencrypted ASPs

2. Create encrypted ASPs

3. Add units to existing ASPs

Selection

F3=Exit F12=Cancel

Selected units have been added successfully

Figure 16-48 IBM i SST Selected units have been added successfully

7. To start IBM i mirrored protection for the system auxiliary storage pool (ASP 1) – or a user ASP 2 to ASP 32 – the IBM i client partition needs to be started to Dedicated Service Tools (DST). This requires a *manual* mode IPL, which we perform by ensuring the IBM i partition properties setting **Keylock position** is set to **manual** on the HMC as shown in Figure 16-49 before exiting from SST and using the PWRDWNSYS RESTART(*YES) CL command to restart the IBM i client partition to DST.

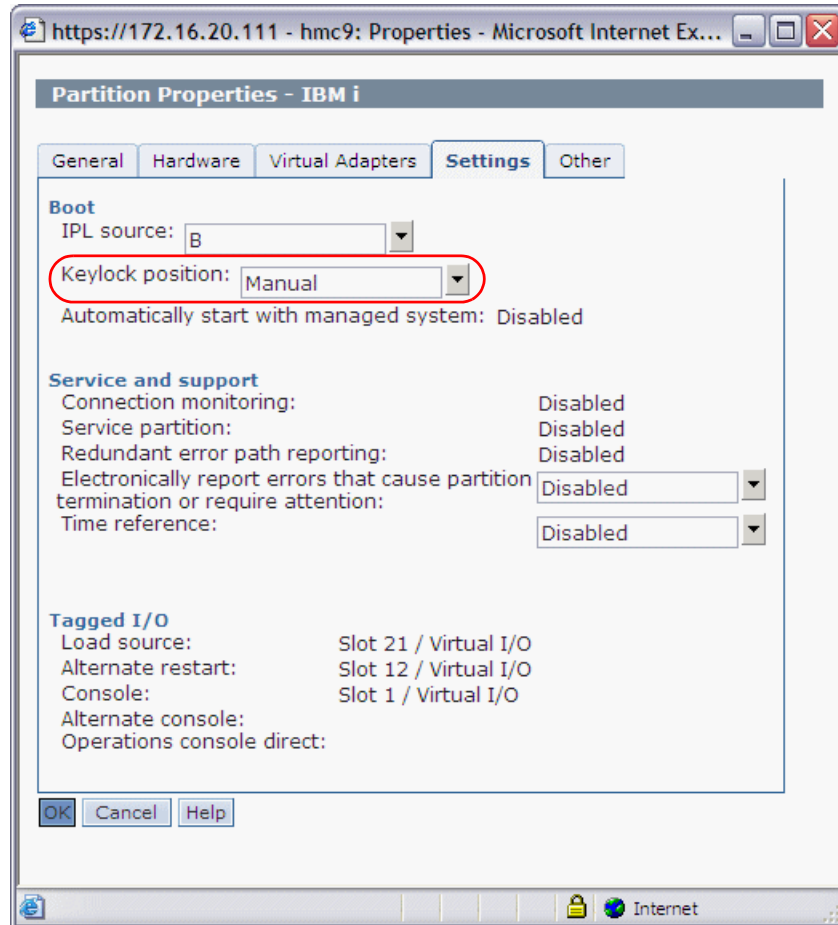


Figure 16-49 IBM i partition restart to DST using a manual IPL

8. After the IBM i partition restart, we are presented with the IPL or Install the System screen, for which we select the option **3. Use Dedicated Service Tools (DST)** to log in to DST and choose the options **4. Work with disk units** → **1. Work with disk configuration** → **4. Work with mirrored protection** → **4. Enable remote load source mirroring** as shown in Figure 16-50. This allows us to mirror the IBM i load source, that is, disk unit 1, across different (virtual) I/O processors (IOPs).

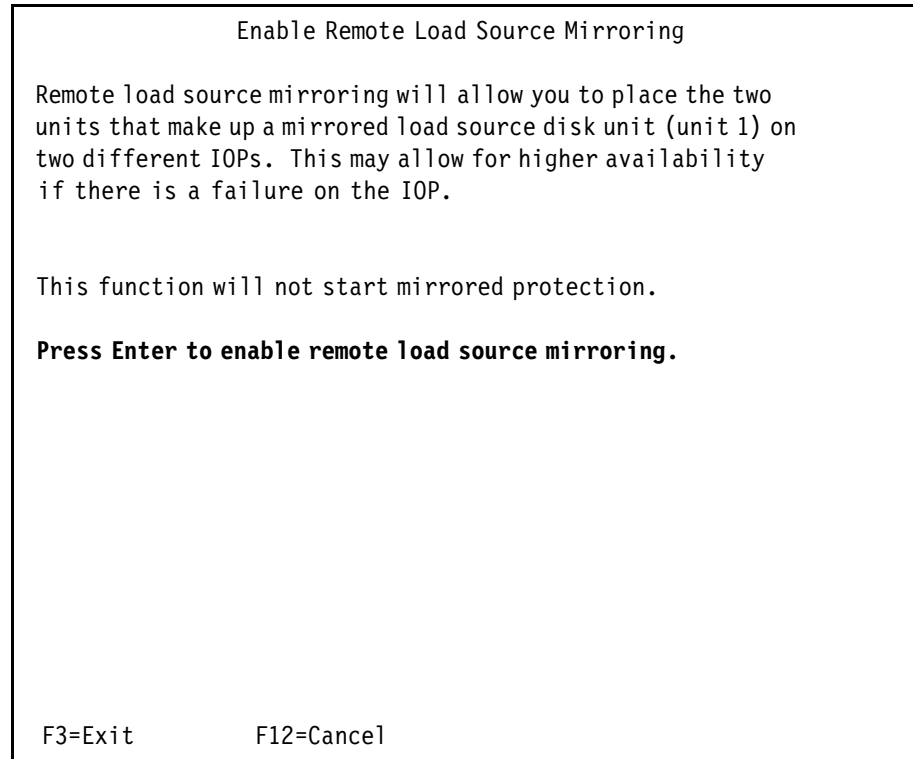


Figure 16-50 IBM i DST Enable remote load source mirroring

9. After pressing Enter to confirm enabling remote load source mirroring, we select option **2. Start mirrored protection** as shown in Figure 16-51.

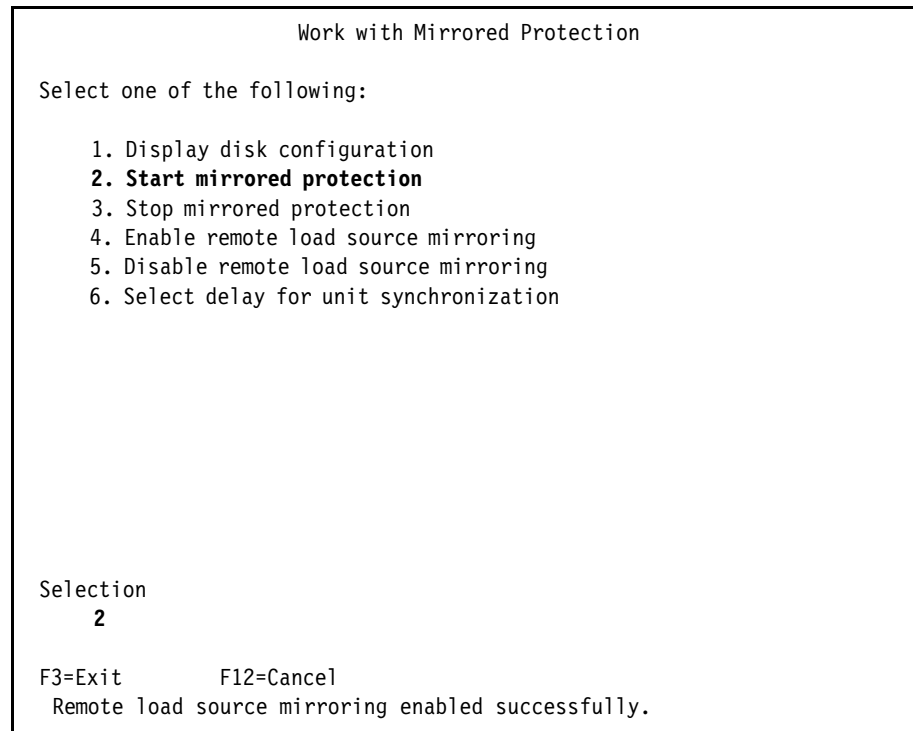


Figure 16-51 IBM i DST Work with mirrored protection

10. We select the ASP which is the system ASP (ASP 1) to start mirrored protection on, as shown in Figure 16-52, and press Enter at the confirmation message to continue.

Select ASP to Start Mirrored Protection

Select the ASPs to start mirrored protection on.

Type options, press Enter.

1=Select

Option	ASP	Protection
1	1	Unprotected

F3=Exit F12=Cancel

Figure 16-52 IBM i DST Select ASP to start mirrored protection

11. At the Problem Report screen for Virtual disk units in the ASP, we select option **5=Display Detailed Report** for demonstration purposes to view an important warning message as shown in Figure 16-53.

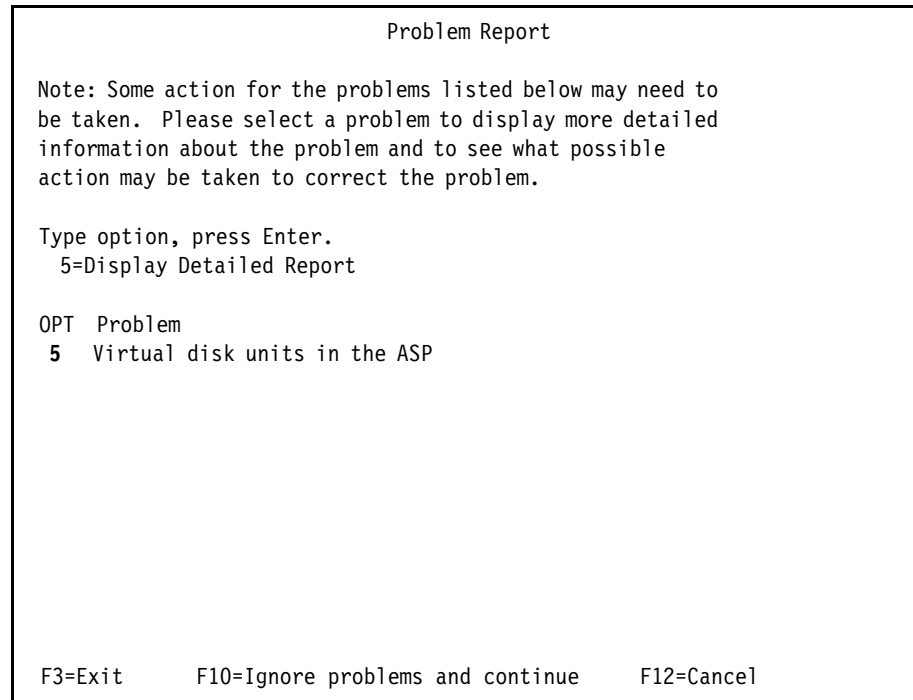


Figure 16-53 IBM i DST Problem Report for Virtual disk units in the ASP

12. We are presented with an important warning message about using mirrored protection with virtual disk units shown in Figure 16-54, which reminds us to make sure that the desired level of mirrored protection, in our case *IOP level mirroring* for mirroring across the two virtual IOPs each connected to another Virtual I/O Server, can be achieved with the virtual I/O configuration, which basically requires the volumes from each mirror side to be provided by another Virtual I/O Server.

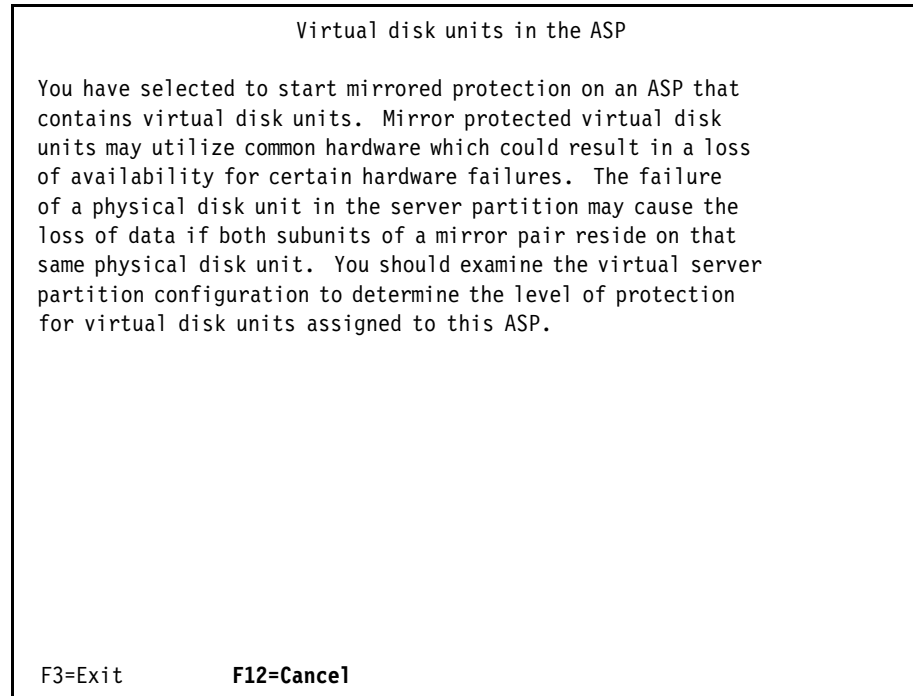


Figure 16-54 IBM i DST Virtual disk units in the ASP message

13. We select **F12=Cancel** to return to the Problem Report screen and acknowledge the potential problem by selecting **F10=Ignore problems and continue**. This allows us to proceed to the Confirm Start Mirrored Protection screen shown in Figure 16-55, which we confirm by pressing Enter.

Confirm Start Mirrored Protection						
Press Enter to confirm your choice to start mirrored protection. During this process the system will be IPLed. You will return to the DST main menu after the IPL is complete. The system will have the displayed protection.						
Press F12 to return to change your choice.						
ASP Unit	Serial Number	Type	Model	Resource Name	Protection	Hot Spare
Protection						
1					Mirrored	
1	Y9UCTLXBVQ9G	6B22	050	DD001	I/O Processor	N
1	Y2LVHS2WFVCM	6B22	050	DD005	I/O Processor	N
F12=Cancel						

Figure 16-55 IBM i DST Confirm Start Mirrored Protection

14. Confirming to start mirrored protection causes the IBM i partition to automatically restart to activate the configuration change as shown in Figure 16-56.

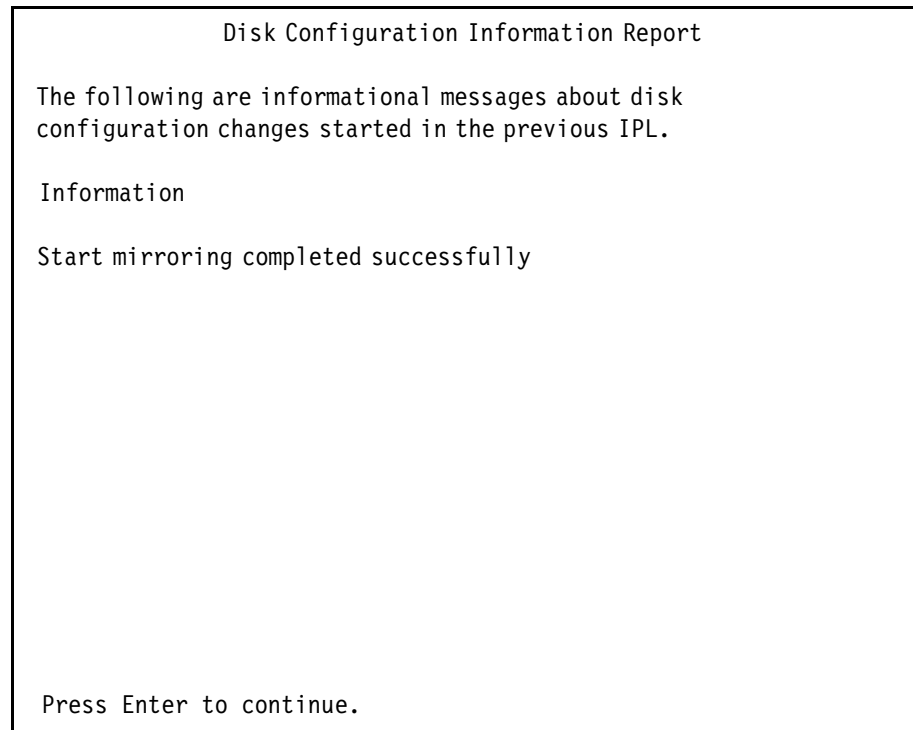


Figure 16-56 IBM i Disk configuration information report

We acknowledge this by pressing Enter to proceed with our manual IPL of the partition, at which the actual mirror synchronization takes place as shown in Figure 16-57.

```

                                Licensed Internal Code IPL in Progress
                                12/07/10  00:42:08

IPL:
Type . . . . . : Attended
Start date and time . . . . . : 12/07/10  00:38:23
Previous system end . . . . . : Abnormal
Current step / total . . . . . :      1    16
Reference code detail . . . . . : C6004050  3

IPL step                                Time Elapsed  Time Remaining
>Storage Management Recovery            00:03:44      00:02:29
Start LIC Log
Main Storage Dump Recovery
Trace Table Initialization
Context Rebuild

Item:
Current / Total . . . . . :      3      10

Sub Item:
Identifier . . . . . : Synchronization of Mirrored Data
Current / Total . . . . . :      60      100
```

Figure 16-57 IBM i Licensed internal code IPL progress

With mirrored protection started now for the system ASP any further disk units can from now on be added to it concurrently – keeping in mind the important notice about bus-level mirroring in step 3 above – using the SST function **3. Work with disk units → 2. Work with disk configuration → 2. Add units to ASPs → 3. Add units to existing ASPs** without requiring another partition restart as shown in Figure 16-58.

Confirm Add Units

Add will take several minutes for each unit. The system will have the displayed protection after the unit(s) are added.

Press Enter to confirm your choice for Add units.
Press F9=Capacity Information to display the resulting capacity.
Press F10=Confirm Add and Balance data on units.
Press F12=Cancel to return and change your choice.

	Serial		Resource		Hot	
Spare						
ASP Unit	Number	Type	Model	Name	Protection	
Protection						
1					Mirrored	
1	Y9UCTLXBVQ9G	6B22	050	DD001	I/O Processor	N
1	Y2LVHS2WFVCM	6B22	050	DD005	I/O Processor	N
2	YYMD6NS9YGL4	6B22	050	DPH002	I/O Processor	N
2	YDU8UT78ZHMZ	6B22	050	DPH005	I/O Processor	N
3	YZG9ZK2YKVV4	6B22	050	DPH001	I/O Processor	N
3	Y6FWEN7UP9DW	6B22	050	DPH004	I/O Processor	N

More...

F9=Resulting Capacity

F10=Add and Balance

F11=Display Encryption Status

F12=Cancel

Figure 16-58 IBM i Confirm Add Units

Our resulting IBM i mirroring disk configuration across two Virtual I/O Servers is shown in Figure 16-59.

Display Disk Configuration Protection						
ASP	Unit	Serial Number	Type	Model	Resource Name	Hot Spare Protection
1					Mirrored	
	1	Y9UCTLXBVQ9G	6B22	050	DD001 I/O Processor	N
	1	Y2LVHS2WFVCM	6B22	050	DD005 I/O Processor	N
	2	YYMD6NS9YGL4	6B22	050	DD009 I/O Processor	N
	2	YDU8UT78ZHMZ	6B22	050	DD010 I/O Processor	N
	3	YZG9ZK2YKVV4	6B22	050	DD007 I/O Processor	N
	3	Y6FWEN7UP9DW	6B22	050	DD006 I/O Processor	N
	4	Y5UQXAAMRRYR	6B22	050	DD011 I/O Processor	N
	4	YY8TMA75JZTR	6B22	050	DD008 I/O Processor	N
Press Enter to continue.						
F3=Exit F5=Refresh F9=Display disk unit details						
F11=Display non-configured units F12=Cancel						

Figure 16-59 IBM i resulting mirroring configuration

Testing mirroring on IBM i client

In the following topic we verify the protection of the IBM i client partition, using mirroring across two Virtual I/O Servers, against Virtual I/O Server outages by simulating a Virtual I/O Server outage with an immediate HMC power-down of the Virtual I/O Server partition.

After the simulated sudden outage of Virtual I/O Server 1, the IBM i client partition loses the mirrored protection with all its disk units from one mirrored side, becoming suspended as reported by a CPI0949 message such as Mirrored protection suspended on disk unit 1. for each affected disk unit, as shown in Figure 16-60.

```
Additional Message Information

Message ID . . . . . : CPI0949      Severity . . . . . : 99
Message type . . . . . : Information
Date sent . . . . . : 12/07/10      Time sent . . . . . : 15:40:07

Message . . . . . : Mirrored protection suspended on disk unit 1.
Cause . . . . . : Mirrored protection is suspended on disk unit number 1.
Data has not been lost. Mirrored protection is suspended for one of the
following reasons:
1. The service representative is repairing the storage unit.
2. A storage unit is not operating.
3. A storage unit found errors.
The following identifies the storage unit.
Disk serial number: Y9UCTLXBVQ9G
Disk type: 6B22
Disk model: 050
Device resource name: DD001

Recovery . . . . . : The system will automatically resume mirroring when the
error has been corrected.
Technical description . . . . . : Error Log ID X'00000000'.

Press Enter to continue.

F3=Exit F6=Print F9=Display message details F12=Cancel
F21=Select assistance level
```

Figure 16-60 IBM i CPI0949 message for a failed disk unit connection

Note that because IBM i mirroring was not suspended manually but by a (forced) error condition, as stated by message CPI0949, the IBM i host will automatically resume mirroring when the error has been corrected.

Displaying the disk configuration status from IBM i System Service Tools again, we can see that all disk units for one mirror side, the one from Virtual I/O Server 1, are in suspended status, while the disk units provided by Virtual I/O Server 2 remain active as shown in Figure 16-61.

Display Disk Configuration Status						
ASP	Unit	Serial Number	Type	Model	Resource Name	Status
					Mirrored	Hot Spare Protection
1	1	Y9UCTLXBVQ9G	6B22	050	DD001	Suspended
	1	Y2LVHS2WFVCM	6B22	050	DD005	Active
	2	YYMD6NS9YGL4	6B22	050	DD009	Suspended
	2	YDU8UT78ZHMZ	6B22	050	DD010	Active
	3	YZG9ZK2YKVV4	6B22	050	DD007	Suspended
	3	Y6FWEN7UP9DW	6B22	050	DD006	Active
	4	Y5UQXAAMRRYR	6B22	050	DD011	Suspended
	4	YY8TMA75JZTR	6B22	050	DD008	Active
Press Enter to continue.						
F3=Exit F5=Refresh F9=Display disk unit details						
F11=Disk configuration capacity F12=Cancel						

Figure 16-61 IBM i SST Display disk path status after outage of Virtual I/O Server1

Note that seeing a dump (SRC B600512D) for any non-load source virtual IOP in the IBM i Product Activity Log when a Virtual I/O Server partition goes away is an expected behavior and nothing to be concerned about.

After Virtual I/O Server 1 is operational again, the IBM i client almost instantly recognizes by its system-wide probes at 15 second intervals that the suspended units from Virtual I/O Server 1 have become operational again, and automatically starts resuming the mirrored disk units as reported by a CPI0988 message such as Mirrored protection resuming on disk unit 1. as shown in Figure 16-62.

```
Additional Message Information

Message ID . . . . . : CPI0988      Severity . . . . . : 40
Message type . . . . . : Information
Date sent . . . . . : 12/07/10      Time sent . . . . . : 16:16:43

Message . . . . . : Mirrored protection resuming on disk unit 1.
Cause . . . . . : Mirrored protection is resuming on disk unit number 1.
One of the steps the system performs before mirrored protection is resumed
is to copy data from the storage unit with serial number Y2LVHS2WFVCM to the
storage unit with serial number Y9UCTLXBVQ9G so that the data on both
storage units is the same. You may observe slower system performance during
the time that the data is being copied. After the copying of the data to
disk is complete, message CPI0989 is sent to this message queue, and
mirrored protection will resume on disk unit number 1.
The following information identifies the resource names for the source
unit:
Device resource name: DD005

The following information identifies the resource names for the target
unit:
Device resource name: DD001

Press Enter to continue.

F3=Exit F6=Print F9=Display message details F12=Cancel
F21=Select assistance level
```

Figure 16-62 IBM i CPI0988 message for resuming mirrored protection

Looking at the disk configuration status in SST again, we can see the progress for resuming mirrored protection as shown in Figure 16-63.

Display Disk Configuration Status						
ASP Unit	Serial Number	Type	Model	Resource Name	Status	Hot Spare Protection
1					Mirrored	
	1 Y9UCTLXBVQ9G	6B22	050	DD001	35 % Resumed	N
	1 Y2LVHS2WFVCM	6B22	050	DD005	Active	N
	2 YYMD6NS9YGL4	6B22	050	DD009	34 % Resumed	N
	2 YDU8UT78ZHMZ	6B22	050	DD010	Active	N
	3 YZG9ZK2YKVV4	6B22	050	DD007	43 % Resumed	N
	3 Y6FWEN7UP9DW	6B22	050	DD006	Active	N
	4 Y5UQXAAMRRYR	6B22	050	DD011	36 % Resumed	N
	4 YY8TMA75JZTR	6B22	050	DD008	Active	N
Press Enter to continue.						
F3=Exit F5=Refresh F9=Display disk unit details						
F11=Disk configuration capacity F12=Cancel						

Figure 16-63 IBM i SST Display disk configuration status for resuming mirroring

After resuming mirrored protection has finished for a mirrored disk unit, a CPI0989 message such as Mirrored protection resumed on disk unit 1. is logged as shown in Figure 16-64.

```
Additional Message Information

Message ID . . . . . : CPI0989      Severity . . . . . : 40
Message type . . . . . : Information
Date sent . . . . . : 12/07/10      Time sent . . . . . : 16:24:05

Message . . . . . : Mirrored protection resumed on disk unit 1.
Cause . . . . . : The system completed the copying of data from the storage
unit with serial number Y2LVHS2WFVCM to the storage unit with serial number
Y9UCTLXBVQ9G. Mirrored protection is resumed on disk unit number 1.
The following information identifies the resource names for the source
unit:
    Device resource name:          DD005
The following information identifies the resource names for the target
unit:
    Device resource name:          DD001

Bottom

Press Enter to continue.

F3=Exit  F6=Print  F9=Display message details  F12=Cancel
F21=Select assistance level
```

Figure 16-64 IBM i CPI0989 message for resumed mirrored protection

Looking at the SST disk configuration status again after resuming mirrored protection has completed successfully for all disk units, we can see all disk units of the mirrored ASP in active status again as shown in Figure 16-65.

Display Disk Configuration Status						
ASP Unit	Serial Number	Type	Model	Resource Name	Status	Hot Spare Protection
1					Mirrored	
	1 Y9UCTLXBVQ9G	6B22	050	DD001	Active	N
	1 Y2LVHS2WFVCM	6B22	050	DD005	Active	N
	2 YYMD6NS9YGL4	6B22	050	DD009	Active	N
	2 YDU8UT78ZHMZ	6B22	050	DD010	Active	N
	3 YZG9ZK2YKVV4	6B22	050	DD007	Active	N
	3 Y6FWEN7UP9DW	6B22	050	DD006	Active	N
	4 Y5UQXAAMRRYR	6B22	050	DD011	Active	N
	4 YY8TMA75JZTR	6B22	050	DD008	Active	N
Press Enter to continue.						
F3=Exit F5=Refresh F9=Display disk unit details						
F11=Disk configuration capacity F12=Cancel						

Figure 16-65 IBM i SST Display disk configuration status after resumed mirroring

Linux client mirroring

In a software RAID mirroring configuration the client partitions are configured with two virtual SCSI adapters. Each of these virtual SCSI adapters is connected to a different Virtual I/O Server and provides one disk to the client partition. On the Virtual I/O Servers, the virtual disks are backed by a logical volume or physical disk. The client partition uses software RAID mirroring so that disk access is redundant. In Linux, **mdadm** provides the software RAID functionality. Figure 16-66 shows the configuration.

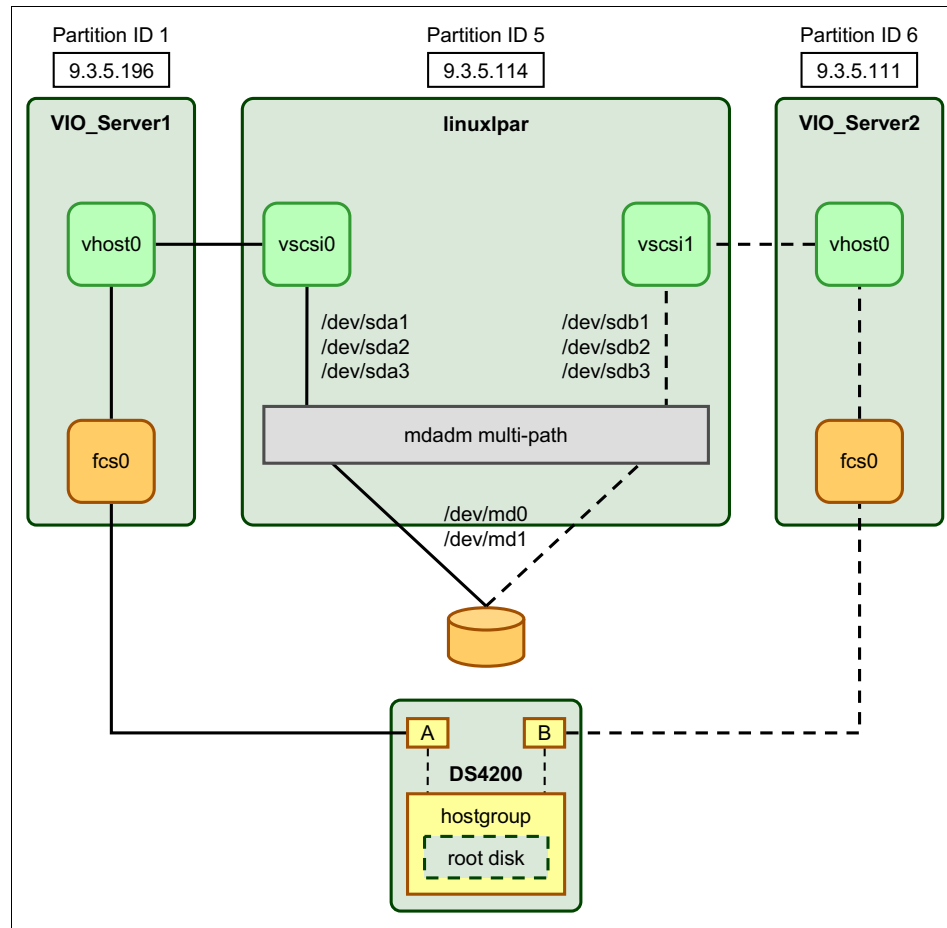


Figure 16-66 Linux client partition using mirroring with mdadm

The first virtual disk (on the virtual adapter with lowest slot number on the client) will become **/dev/sda** in the Linux client partition. The second virtual disk will become **/dev/sdb**.

Figure 16-67 shows a generic disk partitioning configuration for Linux.

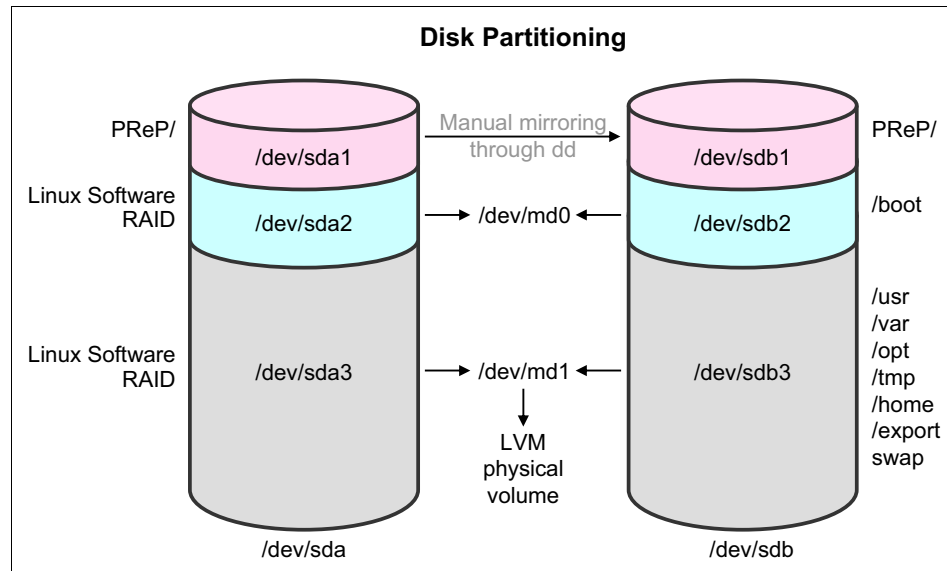


Figure 16-67 Linux partitioning layout for mdadm mirroring

The setup differs depending on the Linux distribution and the version you are using and can be done using the partition tools provided by either the Linux distribution media or IBM Installation Toolkit. Automatic RAID partitioning profiles are also provided by IBM Installation Toolkit. For more information, see this site:

<http://www14.software.ibm.com/webapp/set2/sas/f/lopdiaqs/installtools>

Testing mirroring on the Linux client

If a virtualized disk is not accessible, you will see the disk marked as failed as shown in Example 16-71.

Example 16-71 cat /proc/mdstat showing lost disk

```
[root@op720-1-client4 ~]# cat /proc/mdstat
Personalities : [raid1]
md1 : active raid1 sdb3[1] sda3[0]
      1953728 blocks [2/2] [UU]

md2 : active raid1 sdb4[1] sda4[2] (F)
      21794752 blocks [2/1] [_U]

md0 : active raid1 sdb2[1] sda2[2] (F)
      98240 blocks [2/1] [_U]
unused devices: <none>
```

To recover the disk, perform the following steps:

1. Set the disk to faulty status:

```
# mdadm --manage --set-faulty /dev/md0 /dev/sda2
# mdadm --manage --set-faulty /dev/md1 /dev/sda3
# mdadm --manage --set-faulty /dev/md2 /dev/sda4
```

2. Remove the device:

```
# mdadm --manage --remove /dev/md0 /dev/sda2
# mdadm --manage --remove /dev/md1 /dev/sda3
# mdadm --manage --remove /dev/md2 /dev/sda4
```

3. Rescan the device (choose the corresponding path):

```
echo 1 > /sys/class/scsi_device/0\:0\:1\:0/device/rescan
```

4. Hot add the device to mdadm:

```
# mdadm --manage --add /dev/md0 /dev/sda2
# mdadm --manage --add /dev/md1 /dev/sda3
# mdadm --manage --add /dev/md2 /dev/sda4
```

5. Verify that resynchronization is running:

```
# cat /proc/mdstat
Personalities : [raid1]
md1 : active raid1 sda3[0] sdb3[1]
      1953728 blocks [2/2] [UU]

md2 : active raid1 sda4[2] sdb4[1]
      21794752 blocks [2/1] [_U]
      [=>.....] recovery = 5.8% (1285600/21794752)
      finish=8.2min speed=41470K/sec
md0 : active raid1 sda2[0] sdb2[1]
      98240 blocks [2/2] [UU]
```

Important: Never reboot the other Virtual I/O Server as long as the recovery is not finished.

16.2.8 Shared storage pools

This section gives details on configuration of shared storage pools. We demonstrate creation of shared storage pools, Adding nodes to the cluster, adding physical volumes to the cluster, and creating and mapping logical units from shared storage pools,

Creating a shared storage pool

To create a shared storage pool, you need to create a cluster with a single Virtual I/O Server:

1. Locate the physical volumes to be used for the shared storage pool. For example, use the **lspv -free** command to list physical volumes that are not in use, that is, neither used as a backing device nor as shared memory paging device, as shown in Example 16-72.

Example 16-72 List free physical volumes

```
$ lspv -free
NAME                PVID                SIZE(megabytes)
hdisk0              none                51200
hdisk1              00f61aa6b23980fb    51200
hdisk2              none                51200
hdisk3              none                51200
hdisk4              none                51200
hdisk5              none                51200
hdisk6              none                51200
hdisk7              none                51200
hdisk11             none                140013
$
```

Considerations for creating a shared storage pool:

- ▶ The repository and shared storage pool physical volumes must each have at least 10 GB capacity.
- ▶ The repository and shared storage pool physical volumes must be provided through a SAN.
- ▶ The IP address used for creating the shared storage pool must be the first entry in the `/etc/hosts` file.

2. Create a cluster on the Virtual I/O Server by using the **cluster** command. In our case, we specify `hdisk1` as a repository physical volume and `hdisk2`, `hdisk3` as storage pool physical volumes, as shown in Example 16-73.

Example 16-73 Create a cluster

```
$$ cluster -create -clustername clusterA -repopvs hdisk1 -spname poolA -sppvs hdisk2
hdisk3 -hostname `hostname`
Cluster clusterA has been created successfully.
$
```

Rules: The repository disk belongs to the caavg_private volume group. Volume group commands such as **exportvg** and **lsvg** must not be run on it.

3. Using the **lspv -free** command again to list physical volumes that are not in use after creating the cluster, *hdisk1*, *hdisk2*, and *hdisk3* are removed from the list, because these disks are used now as the repository and storage pool physical volumes as shown in Example 16-74.

Example 16-74 List free physical volumes after creating the cluster

\$ \$ lspv -free		
NAME	PVID	SIZE(megabytes)
hdisk0	none	51200
hdisk4	none	51200
hdisk5	none	51200
hdisk6	none	51200
hdisk7	none	51200
hdisk11	none	140013
\$		

4. To display the physical volumes in the shared storage pool, use the **lspv** command with **clustername** and **sp** flags as shown in Example 16-75.

Example 16-75 List the physical volumes belonging to the shared storage pool

\$ \$ lspv -clustername clusterA -sp poolA		
PV NAME	SIZE(MB)	PVUID
hdisk2	51200	3E213600A0B80001146320000553250893AC30F1815
FASTT03IBMfc		
hdisk3	51200	3E213600A0B8000291B08000036A909AC02580F1815
FASTT03IBMfc		
\$		

Adding nodes

Starting with Virtual I/O Server version 2.2.2.0, you can add up to 16 nodes to the cluster. In Example 16-76, we added 3 nodes to the cluster.

Example 16-76 Adding nodes to a cluster

```
$ cluster -addnode -clustername clusterA -hostname vios02
Partition vios02 has been added to the clusterA cluster.

$ cluster -addnode -clustername clusterA -hostname vios03
Partition vios03 has been added to the clusterA cluster.

$ cluster -addnode -clustername clusterA -hostname vios04
```

Partition `vios04` has been added to the `clusterA` cluster.

\$

To check the status of the cluster and the nodes, use the `cluster` command as shown in Example 16-77.

Example 16-77 Checking the status of the cluster

```
$ cluster -status -clustername clusterA
Cluster Name      State
clusterA          OK

      Node Name      MTM          Partition Num  State  Pool State
      vios01         8233-E8B02061AA6P      1  OK    OK
      vios02         8233-E8B02061AA6P      2  OK    OK
      vios03         8205-E6C0206A22ER      1  OK    OK
      vios04         8205-E6C0206A22ER      2  OK    OK
```

\$

To make sure that a cluster is defined, use the command in Example 16-78 to list the current configuration.

Example 16-78 Listing the cluster information

```
$ cluster -list
CLUSTER_NAME:    clusterA
CLUSTER_ID:      9c189d6c286711e2b8af00145ee9e161
```

\$

Adding physical volumes to the shared storage pool

Before you start adding additional physical volumes to the cluster, ensure that there are valid candidates for being part of the shared storage pool. Example 16-79 shows how to display a list of physical volumes capable of being added.

Example 16-79 List of physical volumes capable of being added

```
$ lspv -clustername clusterA -capable
PV NAME      SIZE(MB)  PVUID
hdisk0       51200    3E213600A0B80001146320000552F50893A8E0F1815
FAStT03IBMfc
hdisk4       51200    3E213600A0B80001146320000553450893B870F1815
FAStT03IBMfc
```

```

hdisk5          51200      3E213600A0B8000291B08000036AB09AC02890F1815
FAStT03IBMfcp
hdisk6          51200      3E213600A0B80001146320000553650893BB80F1815
FAStT03IBMfcp
hdisk7          51200      3E213600A0B8000291B08000036AD09AC02BC0F1815
FAStT03IBMfcp
$

```

To add a physical volume to the shared storage pool, use the **chsp** command as shown in Example 16-80.

Example 16-80 Adding the physical volume to the shared storage pool

```

$ chsp -add -clustername clusterA -sp poolA hdisk0
Current request action progress: % 5
Current request action progress: % 5
Current request action progress: % 80
Current request action progress: % 100
$

```

To display the physical volumes in the shared storage pool, use the **lspv** command as shown in Example 16-81.

Example 16-81 A list of the physical volumes in the shared storage pool

```

$ lspv -clustername clusterA -sp poolA
PV NAME          SIZE(MB)    PVUIDID
hdisk2           51200      3E213600A0B80001146320000553250893AC30F1815
FAStT03IBMfcp
hdisk3           51200      3E213600A0B8000291B08000036A909AC02580F1815
FAStT03IBMfcp
hdisk0           51200      3E213600A0B80001146320000552F50893A8E0F1815
FAStT03IBMfcp
$

```

To display the size of the shared storage pool and free space, use the **lssp** command as shown in Example 16-82.

Example 16-82 Listing the shared storage pool

```

$ lssp -clustername clusterA
POOL_NAME:      poolA
POOL_SIZE:      153216
FREE_SPACE:     150562
TOTAL_LU_SIZE:  100
OVERCOMMIT_SIZE: 0
TOTAL_LUS:      1

```

POOL_TYPE: CLPOOL
POOL_ID: FFFFFFFFAC10156F00000000509999C1

\$

Create and map logical units in a shared storage pool

This section shows how to create shared storage pool backed logical units and map them to a client partition.

Two types of logical units can be created in a shared storage pool, thin and thick. The default logical unit is thin, meaning it will only use a minimal initial space on the physical disk and it will not significantly reduce the size of the pool. In case of a thick unit the actual size of the logical unit will be allocated on the physical disks from the shared storage pool and this will be reflected when checking the size of the pool. Thick provisioning is similar to taking a slice from the storage subsystem and assigning it to a virtual server.

Logical units from a shared storage pool can be assigned to AIX, IBM i and Linux partitions.

Use the `lsmap -all` command to display the physical location for the virtual SCSI server adapter as shown in Example 16-83.

Example 16-83 List the physical location of the virtual SCSI server adapter

```
$ $ lsmap -all
SVSA          Physloc                      Client Partition ID
-----
vhost0        U8233.E8B.061AA6P-V1-C5      0x00000004

VTD           NO VIRTUAL TARGET DEVICE FOUND

SVSA          Physloc                      Client Partition ID
-----
vhost1        U8233.E8B.061AA6P-V1-C13     0x00000003

VTD           NO VIRTUAL TARGET DEVICE FOUND

SVSA          Physloc                      Client Partition ID
-----
vhost2        U8233.E8B.061AA6P-V1-C14     0x00000004

VTD           NO VIRTUAL TARGET DEVICE FOUND

SVSA          Physloc                      Client Partition ID
```

```
-----  
vhost3          U8233.E8B.061AA6P-V1-C55          0x00000005  
  
VTD              NO VIRTUAL TARGET DEVICE FOUND  
  
$  
-----
```

5. Create the logical unit by using the **mkbdsp** command. In Example 16-84, the logical unit *luA1* is created with a initial provisional size of 100 MB.

Example 16-84 create the shared storage pool backed logical unit

```
$ $ mkbdsp -clustername clusterA -sp poolA 100M -bd luA1  
Lu Name:luA1  
Lu Udid:ed0dd2e296eb187e940fbaf432141756
```

```
$  
-----
```

- iii. Map the logical unit created previous step to the virtual SCSI server adapter associated with a client partition by using the **mkbdsp** command as shown in Example 16-85.

Example 16-85 Map the logical unit to the virtual SCSI server adapter

```
$ $ mkbdsp -clustername clusterA -sp poolA -bd luA1 -vadapter vhost0  
Assigning file "luA1" as a backing device.  
VTD:vtscsi0
```

```
$  
-----
```

Tips:

- ▶ Creating and mapping the logical unit can also be done simultaneously by specifying the size of the logical unit and using the **-vadapter** flag.
- ▶ An arbitrary virtual target device name can be specified by using the **-tn** flag in **mkbdsp** command.

6. Display the logical unit information by using the **lssp** command and **lsmmap -all** command as shown in Example 16-86.

Example 16-86 List the logical unit information

\$ lsmmap -all		
SVSA ID	Physloc	Client Partition

vhost0	U8233.E8B.061AA6P-V1-C5	0x00000004
VTD	vtscsi0	
Status	Available	
LUN	0x8100000000000000	
Backing device	1uA1.ed0dd2e296eb187e940fbaf432141756	
Physloc		
Mirrored	N/A	
SVSA ID	Physloc	Client Partition

vhost1	U8233.E8B.061AA6P-V1-C13	0x00000003
VTD	NO VIRTUAL TARGET DEVICE FOUND	
SVSA ID	Physloc	Client Partition

vhost2	U8233.E8B.061AA6P-V1-C14	0x00000004
VTD	NO VIRTUAL TARGET DEVICE FOUND	
SVSA ID	Physloc	Client Partition

vhost3	U8233.E8B.061AA6P-V1-C55	0x00000005
VTD	NO VIRTUAL TARGET DEVICE FOUND	
\$		

7. On an AIX client partition, use the **cfgmgr** command to make the virtual physical volume mapped in the previous step visible.

Shared storage pools can only contain disks from a SAN subsystem with no restrictions as long as the devices are supported by the multipath driver on the Virtual I/O Server. If you need to increase the free space in the shared storage pool, you can either add an additional physical volume or you can replace an existing volume with a bigger one.

Physical disks cannot be removed from the shared storage pool. More about replacing a physical volume in the cluster can be found in *IBM PowerVM Virtualization Managing and Monitoring*, SG24-7590.

When using thin provisioned devices the total size of logical units can be larger than the size of the shared storage pool. However, the free space of the shared storage pool becomes small if the actual physical usage of logical units becomes larger. In this case you need to add an additional physical volume to the shared storage pool.

To create a thick unit, specify the **-thick** parameter for mkbdsp as shown in Example 16-87.

Example 16-87 Creating a thick logical unit

\$ lssp -clustername clusterA -sp poolA -bd					
Lu Name	Size(mb)	ProvisionType	%Used	Unused(mb)	Lu Udid
luA	10240	THIN	0%	10240	
8655f81299fd6f4f004a627667653975					
luA	10240	THICK	100%	0	
e352a6b3ea352caace0221e8ea43f0a6					
\$					

Tip: Logical unit names do not have to be unique, however it makes administration easier. In case of unique names, you have to use the Lu Udid to identify the logical unit.

To map the logical unit, use the same command as shown in Example 16-88. Notice the usage of the **-luudid** parameter.

Example 16-88 Mapping the logical unit to a vhost adapter

\$ mkbdsp -clustername clusterA -sp poolA -bd luA -vadapter vhost0	
Specified LU is not unique. Please select the LU UDID from the below list.	
LU Name	LU UDID
luA	e352a6b3ea352caace0221e8ea43f0a6
luA	8655f81299fd6f4f004a627667653975
\$ mkbdsp -clustername clusterA -sp poolA -luudid	
8655f81299fd6f4f004a627667653975 -vadapter vhost0	
Assigning file "8655f81299fd6f4f004a627667653975" as a backing device.	
VTD:vtscsi0	
\$	

You can also create and assign the logical unit in one command as shown in Example 16-89.

Example 16-89 Creating and mapping of a logical unit with one command

```
$$ mkbdsp -clustername clusterA -sp poolA 10G -bd luB -vadapter vhost0
Lu Name:luB
Lu Udid:9550b5c97681fedd52e1f0cf15b38dad

Assigning file "luB" as a backing device.
VTD:vtscsi1

$
```

After mapping is complete you should discover the disk on the client partition and check that it's parameters are correct. Make sure you pay attention to the *queue_depth* value. The default of 3 can be safely changed as usually there are several physical backing devices behind every virtual disk in a shared storage pool. Example 16-90 shows the default value.

Example 16-90 Listing the attributes of a disk

```
root@p71aix90 /tmp/sysdir/agents # lsattr -El hdisk1
```

PCM	PCM/friend/vscsi	Path Control Module	False
algorithm	fail_over	Algorithm	True
hcheck_cmd	test_unit_rdy	Health Check Command	True
hcheck_interval	0	Health Check Interval	True
hcheck_mode	nonactive	Health Check Mode	True
max_transfer	0x40000	Maximum TRANSFER Size	True
pvid	00f70ef5fbfb8bd40000	Physical volume identifier	False
queue_depth	3	Queue DEPTH	True
reserve_policy	no_reserve	Reserve Policy	True

The recommended starting value for *queue_depth* is 32 and can be increased in case the *avg_wqsz* size as reported by **iosstat -D** command is consistently above 0. To change the value use **chdev -l hdisk1 -a queue_depth=32 -P** and restart the system. The value cannot be changed while the disk is in use.

Remember: The SCSI limitation is 512 command elements for each vscsi adapter out of which 2 are reserved for the adapter and 3 for each vdisk. You can use this formula for calculations of queue depth:
virtual_disks=(512-2)/(Q+3) where **Q** equals the *queue_depth* of each virtual disk.

16.3 Network virtualization setup

This section presents the most common high availability network virtualization scenarios:

- ▶ Multiple VLANs
- ▶ SEA failover
- ▶ EtherChannel Backup in the AIX client
- ▶ Linux Ethernet connection bonding
- ▶ General rules for setting modes for QoS
- ▶ Denial of Service hardening

16.3.1 Multiple VLANs

In a PowerVM environment, it is the Virtual I/O Server that provides the link between the internal virtual and external physical LANs. This can introduce an increased level of complexity due to multiple VLANs within a server that needs to be connected to multiple VLANs outside the server in a secure manner. The Virtual I/O Server, therefore, needs to be connected to all of the VLANs but must not allow packets to move between the VLANs.

In this scenario, the following requirements must be met:

- ▶ All client partitions must be able to communicate with other client partitions on the same VLAN.
- ▶ All client partitions must be able to communicate to a single virtual Ethernet adapter in the Virtual I/O Server. This is achieved by using IEEE 802.1Q on the Virtual I/O Servers virtual Ethernet adapter to allow more than one VLAN ID to be accepted at the virtual Ethernet adapter.
- ▶ VLAN tags must not be stripped from arriving packets from the client partitions. Stripped VLAN tags causes Virtual I/O Server not being able to forward the packets to the correct external VLAN.
- ▶ Shared Ethernet Adapter (SEA) must be enabled to allow packets from multiple VLANs.

This implementation requires that the enterprise security policy encompasses the following considerations:

- ▶ The enterprise security policy recognizes IEEE 802.1Q VLAN tagging.

The IEEE 802.1Q VLAN tagging is implemented in the PowerVM hypervisor firmware. The Virtual I/O Server is able to have up to 21 VLANs per Shared Ethernet Adapter, but in order to use these, the physical network port must support the same number of VLANs. The physical VLAN policy within the enterprise therefore determines the virtual VLAN policy.

- ▶ The enterprise security policy allows a network switch to have multiple VLANs.

The enterprise security policy allows multiple VLANs to share a network switch (non-physical security). If it is a security requirement that a network switch only have one VLAN, then every VLAN requires a separate Shared Ethernet Adapter or Virtual I/O Server. If you just make a separate Virtual I/O Server in a managed system, the hypervisor firmware acts like one switch with multiple VLANs, which in this case, is not allowed by the security policy outside the Virtual I/O Server.

Attention: Security in a virtual environment depends on the HMC or Integrated Virtualization Manager (IVM) and the Virtual I/O Server. Access to the HMC, IVM, and Virtual I/O Server must be closely monitored to prevent unauthorized modification of existing network and VLAN assignments, or establishing new network assignments on LPARs within the managed systems.

Figure 16-68 shows four partitions and a single Virtual I/O Server. In this example the VLAN interfaces are defined on an SEA adapter so that the Virtual I/O Server can communicate on these VLANs.

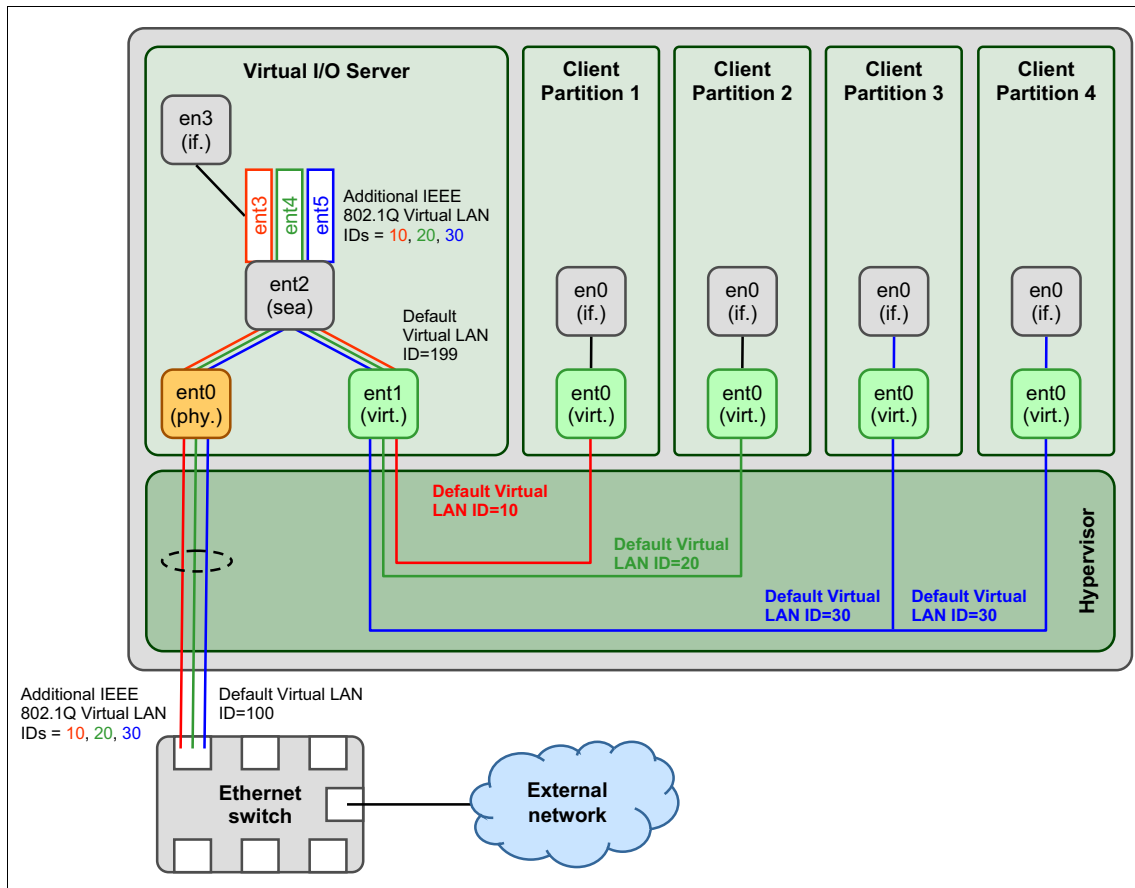


Figure 16-68 VLAN configuration scenario

The following VLAN IDs 10, 20, and 30 are used. In addition, the default VLAN ID 199 is also used. This default VLAN ID must be unique and not used by any clients in the network or physical Ethernet switch ports, as in Figure 16-68. For further details, see “Ensuring VLAN tags are not stripped on the Virtual I/O Server” on page 585.

Support: Not all physical network switches support VLAN tagging. If you want to extend VLAN tagging outside your virtual network to your physical network, your physical network must also support VLAN tagging.

Configuring the client partitions

The internal VLAN setup is a straightforward process. Perform the following steps on each client partition to configure the internal VLAN:

1. Log on to the HMC.
2. Start creating a partition.
3. In the virtual Ethernet adapter window, assign each client partition a virtual Ethernet adapter with a default VLAN ID, as shown in Figure 16-68 on page 582. Make sure both the Access external network and IEEE 802.1Q compatible adapter flags are disabled. Figure 16-69 here shows an example of creating a virtual Ethernet adapter for client partition 1.

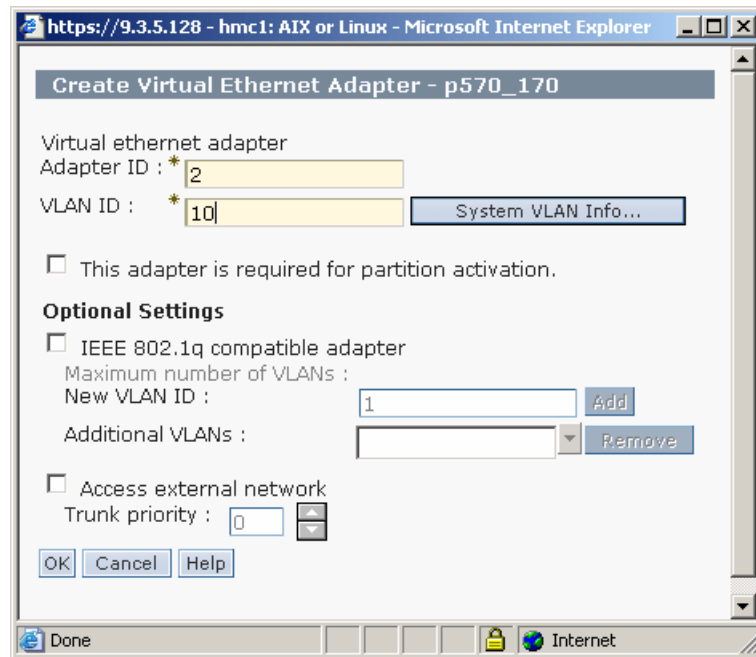


Figure 16-69 Virtual Ethernet configuration for the client partition using the HMC

4. Finalize the partition creation.

After the configuration completes, the VLAN ID on the virtual Ethernet adapter is added to the packets from the client partition. This ID is used to route packets to the correct partition. The Hypervisor will strip them off before delivery. For example, as shown in Figure 16-68 on page 582, client partition 3 and client partition 4 have the same VLAN ID, and thus are able to communicate directly. Client partition 1 and client partition 2 have different VLAN IDs, and thus are unable to communicate with client partition 3 and client partition 4 or with each other, but are ready to communicate with other machines on VLANs external to the machine.

Configuring the Virtual I/O Server

The Virtual I/O Server is the relay point between multiple internal VLANs and the external physical Ethernet adapter or adapters and LAN. To bridge the virtual and physical networks, the Shared Ethernet Adapter (SEA) device is used to link one or more virtual adapters to the physical adapter. Using the scenario in Figure 16-68 on page 582, we implement a single virtual Ethernet adapter connecting to multiple internal VLANs.

To correctly set up this configuration, the virtual Ethernet adapter that is created in the Virtual I/O Server needs to name all the internal VLANs that are connected. Perform the following steps:

1. Log on to the HMC.
2. In the virtual Ethernet adapter window, assign the Virtual I/O Server a virtual Ethernet adapter with a unique default VLAN ID. This ID must not be used by any client partition or physical network. In this scenario, we use the VLAN ID 199.
3. Select the **IEEE 802.1Q compatible adapter** flag.
4. Add additional VLAN IDs associated with the client partitions. In this scenario, the additional VLAN IDs are 10, 20, 30. To display a list of the VLANs which have been added click the arrow next to the **additional VLANs** field.
5. Make sure that the **Access external network** flag is selected. Leave the Trunk priority as the default unless using a SEA failover configuration.

Figure 16-70 shows the setup for this configuration.

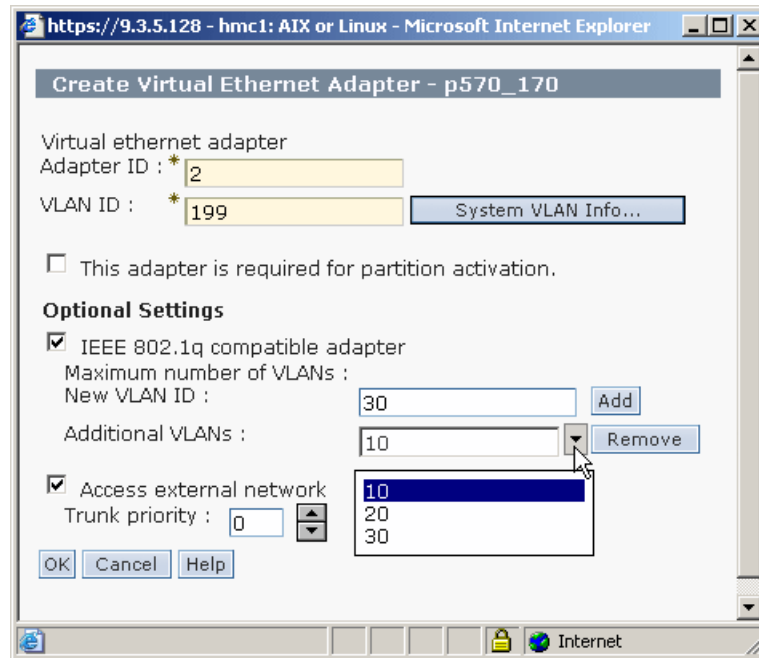


Figure 16-70 Virtual Ethernet configuration for Virtual I/O Server using the HMC

In this instance, the **Access external network** flag is selected because this Virtual I/O Server is using the Shared Ethernet Adapter function to transfer packets to and from the external network. The **IEEE 802.1Q compatible adapter** flag is selected to allow the Shared Ethernet Adapter to transmit packets on additional VLAN IDs. These additional VLAN IDs are the same as the VLAN IDs configured in the client partitions; therefore, packets that arrive from the client partition with the VLAN tag added by the hypervisor are allowed to pass through the Shared Ethernet Adapter.

Ensuring VLAN tags are not stripped on the Virtual I/O Server

It is important that the VLAN ID added to the packets leaving the client partitions (the default VLAN ID number) is not removed on entering the Virtual I/O Server. This is what happens when a default configuration is used.

It is for this reason that the default VLAN ID on the Virtual I/O Servers virtual Ethernet adapter, as shown in Figure 16-70, must be set to an unused VLAN ID (in this scenario, 199).

If a packet arrives with this VLAN ID (199), the VLAN ID tag will be stripped off (untagged) and cannot then be forwarded on to the correct external VLAN. If this was sent through the Shared Ethernet Adapter to the external physical network, it will arrive at the Ethernet switch as untagged. The resulting outcome depends on the settings on the physical Ethernet switch. The packet might be discarded or sent on to a default VLAN, but the chances of it going to VLAN ID 199 are remote unless the network administrator has explicitly set this up (for example, the Ethernet switch port has a default VLAN ID of 199).

Extending multiple VLANs into client partitions

In a client partition, VLANs may be configured in two different ways. Figure 16-71 shows a configuration where both Partition 1 and Partition 2 can communicate on two different VLANs.

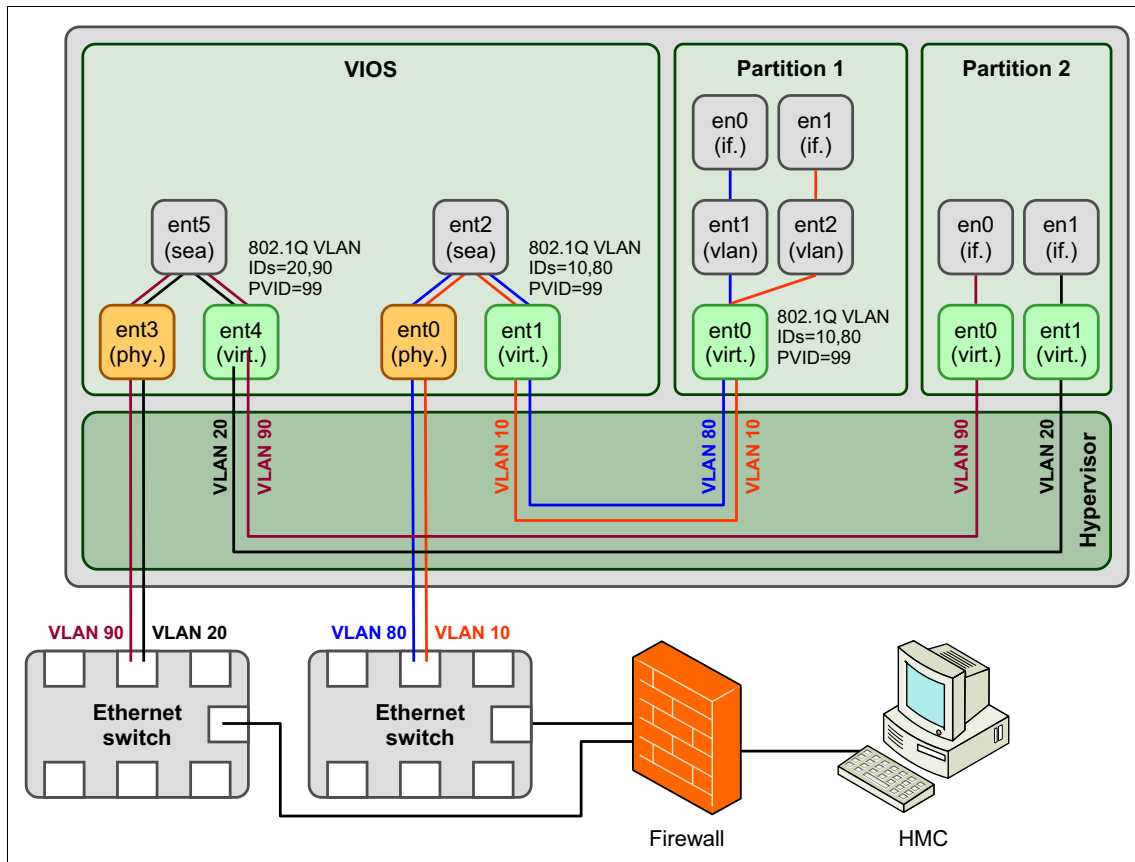


Figure 16-71 HMC in a VLAN tagged environment

Partition 1 has only one Virtual Ethernet adapter that has an unused PVID configured and VLANs 10 and 80 configured as 802.1Q VLANs. For VLAN 10 and 80 there is a separate network interface configured on the Virtual Ethernet adapter. A network packet sent through en0 is tagged with VLAN ID 80 in the client partition. No VLAN tag is added by the Hypervisor switch. The VLAN tag is not stripped off by the SEA because the VLAN ID does not match the PVID of the virtual Ethernet adapter in the SEA.

Partition 2 has two virtual Ethernet adapters where the VLAN is configured using the PVID. One is configured for VLAN 20 and the other for VLAN 90. No VLAN interfaces need to be configured in the client partition. A network packet sent through en0 is tagged with VLAN ID 90 by the Hypervisor switch. The VLAN tag is not stripped off by the SEA because the VLAN ID does not match the PVID of the virtual Ethernet adapter in the SEA. This method is normally the preferred one because it is easier to configure in the client partition.

For dynamic LPAR operations the HMC needs connectivity to all the partitions. Therefore it is important to consider in which network the HMC is placed. Because the HMC does not offer VLAN tagging, a solution can be to put the HMC behind a firewall as shown in Figure 16-71 on page 586 and Figure 16-72.

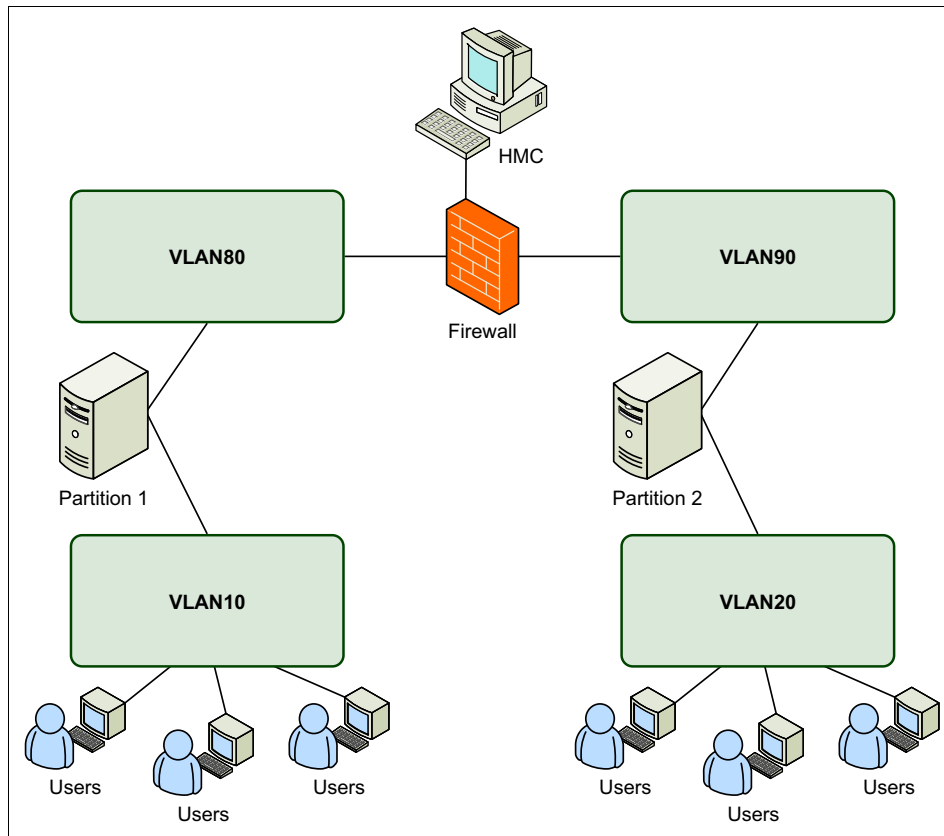


Figure 16-72 Cross-network VLAN tagging with a single HMC

Shared Ethernet Adapter for VLAN use

Based on this scenario, to configure the Shared Ethernet Adapter and access the multiple VLANs, perform the following steps:

1. Use the **mkvdev -sea** command on the Virtual I/O Server to create the Shared Ethernet Adapter. In this scenario, the virtual Ethernet adapter ent1 and physical Ethernet adapter ent0 are used to create the new Shared Ethernet Adapter ent2.

```
$ mkvdev -sea ent0 -vadapter ent1 -default ent1 -defaultid 199
ent2 Available
ent
et2
```

When creating the Shared Ethernet Adapter, the `-default ent1` option is the default internal virtual Ethernet adapter to send the packet onto if the packet is untagged. In this scenario, there is only one virtual Ethernet adapter, but this will be used in the more complex case of multiple virtual Ethernet adapters specified in the command. Also, the `-defaultid 199` option is the VLAN ID to use for untagged packets (in effect, this is the default VLAN ID of the SEA). We are not expecting untagged packets, and by using an unused number like this, these packets are not delivered because no client will accept VLAN 199 packets. Running the `lsdev` command displays the newly created SEA:

```
$ lsdev -type adapter
name          status
description
ent0           Available  10/100/1000 Base-TX PCI-X Adapter
(14106902)
ent1           Available  Virtual I/O Ethernet Adapter (l-lan)
ent2           Available  Shared Ethernet Adapter
ide0           Available  ATA/IDE Controller Device
sisioa0        Available  PCI-X Dual Channel U320 SCSI RAID Adapter
vhost0         Available  Virtual SCSI Server Adapter
vsa0           Available  LPAR Virtual Serial Adapter
```

2. Using the `mkvdev -vlan` command, configure the newly created Shared Ethernet Adapter (`ent2`) to allow packets from the additional VLAN IDs (10, 20, 30):

```
$ mkvdev -vlan ent2 -tagid 10
ent3 Available
en3
et3
$ mkvdev -vlan ent2 -tagid 20
ent4 Available
en4
et4
$ mkvdev -vlan ent2 -tagid 30
ent5 Available
en5
et5
```

Tip: This creates a new Ethernet adapter for each of the previous commands. The addition of VLAN interfaces to the SEA adapter is only necessary if the VIO Server itself needs to communicate on these VLANs.

3. Using the **mktcpip** command or the **cfgassist** command, configure an IP address on one of the new VLANs to allow administrators network access to the Virtual I/O Server. In this scenario, as shown in Figure 16-68 on page 582, we use en3, which resides on VLAN 10, to configure an IP address for Virtual I/O Server network connectivity.

Tip: Typically, the IP address is only required on the Virtual I/O Server for administrative purposes. As such, you can configure the IP address on the management VLAN only to allow administrators network access to the Virtual I/O Server.

Virtual Ethernet and SEA considerations

The following considerations apply when implementing virtual Ethernet and Shared Ethernet Adapters in the Virtual I/O Server:

- ▶ Virtual Ethernet requires a POWER5 or later processor-based system running IVM or an HMC to define the virtual Ethernet adapters.
- ▶ Virtual Ethernet is available on all POWER5 or later processor-based systems, while Shared Ethernet Adapter and the Virtual I/O Server require the configuration of the PowerVM Standard Edition or PowerVM Enterprise Edition feature on some models.
- ▶ Virtual Ethernet can be used in both shared and dedicated processor partitions.
- ▶ Virtual Ethernet does not require a Virtual I/O Server for communication between partitions in the same system.
- ▶ A maximum of up to 256 virtual Ethernet adapters are permitted per partition.
- ▶ Each virtual Ethernet adapter is capable of being associated with up to 20 VLANs (19 VIDs and 1 PVID).
- ▶ A system can support up to 4096 different VLANs, as defined in the IEEE802.1Q standard.
- ▶ The partition must be running a minimum level of AIX Version 5.3, IBM i 6.1, or Linux with the 2.6 kernel or a kernel that supports virtual Ethernet.

- ▶ A mixture of virtual Ethernet connections, real network adapters, or both are permitted within a partition.
- ▶ Virtual Ethernet can only connect partitions within a single system.
- ▶ Virtual Ethernet connections between AIX, IBM i and Linux partitions are supported.
- ▶ Virtual Ethernet uses the system processors for all communication functions instead of off-loading that load to processors on network adapter cards. As a result, there is an increase in system processor load generated by the use of virtual Ethernet.
- ▶ Up to 16 virtual Ethernet adapters with 20 VLANs (19 VID and 1 PVID) on each can be associated to a Shared Ethernet Adapter, sharing a single physical network adapter.
- ▶ There is no explicit limit on the number of partitions that can attach to a VLAN. In practice, the amount of network traffic will limit the number of clients that can be served through a single adapter.
- ▶ To provide highly available virtual Ethernet connections to external networks, two Virtual I/O Servers with Shared Ethernet Adapter Failover or another network HA mechanism has to be implemented.
- ▶ You cannot use SEA failover with Integrated Virtualization Manager (IVM), because IVM only supports a single Virtual I/O Server.

16.3.2 SEA failover

In the SEA failover scenario, the Shared Ethernet Adapters communicate with each other on a *control channel* using two virtual Ethernet adapters configured on a separate VLAN. The control channel is used to carry heartbeat packets between the two Shared Ethernet Adapters. When the primary Shared Ethernet Adapter loses connectivity or the Virtual I/O Server is shut down for maintenance, the network traffic is automatically switched to the backup Shared Ethernet Adapter.

Figure 16-73 shows the configuration. For the sake of simplicity, only the AIX client partition DB_Server is shown in the picture. In a Shared Ethernet Adapter failover configuration, there are two Virtual I/O Servers, each running a Shared Ethernet Adapter.

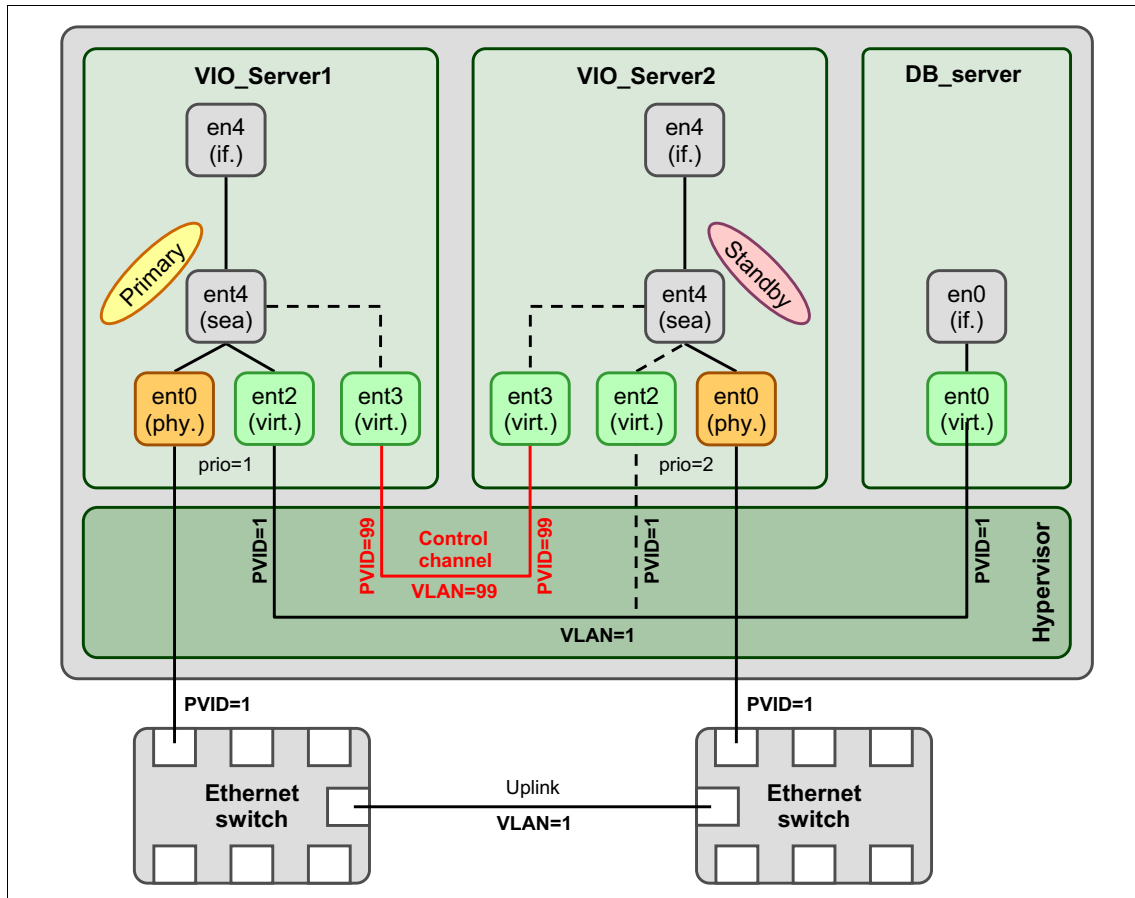


Figure 16-73 Highly available Shared Ethernet Adapter setup

Shared Ethernet Adapter failover works in the same fashion for AIX, IBM i, or Linux client partitions. No special configuration is required in the client partition.

In most installation scenarios, Shared Ethernet Adapter failover is the best solution for providing redundant access to external networks because it is easy to set up and maintain.

Ethernet connection bonding in an AIX or Linux client partition offers more flexibility for separating network traffic between the Virtual I/O Servers but requires more configuration efforts.

Configuring SEA failover

Use the following steps to set up the scenario:

1. Create two Virtual I/O Server partitions and name them VIO_Server1 and VIO_Server2, following the instructions in “Creating a Virtual I/O Server” on page 312. In step 10, select one physical Ethernet adapter and one physical storage adapter.
2. Install both Virtual I/O Servers by following the instructions in “Installation of Virtual I/O Server” on page 333.
3. Configure each Virtual I/O Server with two virtual Ethernet adapters. One virtual Ethernet adapter is configured on VLAN 1 and is used for network traffic to the external network. The second virtual Ethernet adapter is on VLAN 99 and is used for heartbeat traffic between to the SEAs. Table 16-8 shows an overview of the required virtual Ethernet adapters. The *trunk priority* for the Virtual Ethernet adapter on VIO_Server1 which has the Access external network flag set is set to 1. This means that normally the network traffic will go through VIO_Server1. VIO_Server2 with trunk priority 2 is used as backup in case VIO_Server1 has no connectivity to the external network.

Table 16-8 Virtual Ethernet adapter overview for Virtual I/O Servers

Virtual I/O Server	Virtual I/O Server slot	VLAN ID	Trunk priority	Access external network
VIO_Server1	10	1	1	yes
VIO_Server1	11	99	0	no
VIO_Server2	10	1	2	yes
VIO_Server2	11	99	0	no

Tip: The two adapters used for SEA heartbeat must be on the same VLAN. The VLAN must be the same for both adapters and must not be used by any other adapter on the machine.

4. Also make the virtual SCSI or virtual Fibre Channel configuration highly available by either configuring multipathing as described in “Availability configurations using multipathing” on page 501 or mirroring as described in “Availability configurations using mirroring” on page 534.

5. Define the SEA adapter device on VIO_Server1 and VIO_Server2 using the **mkvdev** command, as shown in Example 16-91. The SEA is defined with an IP address to ping using the **-netaddr** flag. The SEA will periodically ping this IP address, so it can detect some other network failures. This is similar to the IP address to ping that can be configured with Network Interface Backup. In this example, the SEA is also configured with the *largesend* option to enable TCP segmentation offload.

Example 16-91 Shared Ethernet Adapter with failover creation

```
$ mkvdev -sea ent0 -vadapter ent2 -default ent2 -defaultid 1 -attr
ha_mode=auto ctl_chan=ent3 netaddr=9.3.4.1 largesend=1
ent4 Available
en4
et4
```

Important: Mismatching SEA and SEA failover can cause broadcast storms to occur and affect the stability of your network. When upgrading from an SEA to an SEA failover environment, it is imperative that the Virtual I/O Server with the regular SEA is modified to SEA failover *prior* to creating the second SEA with SEA failover enablement.

Tip: If you have an existing SEA, to configure it to failover mode, you use the **chdev** command as shown here:

```
$ chdev -dev ent4 -attr ha_mode=auto ctl_chan=ent3
```

6. Verify the SEA adapter attributes on both Virtual I/O Servers' SEA adapters, as shown in Example 16-92.

Example 16-92 Verify and change attributes for SEA adapter

```
$ lsdev -dev ent4 -attr
attribute      value      description
user_settable

ctl_chan       ent3       Control Channel adapter for SEA failover                True
gvrp           no         Enable GARP VLAN Registration Protocol (GVRP)           True
ha_mode        auto       High Availability Mode                                   True
jumbo_frames   no         Enable Gigabit Ethernet Jumbo Frames                    True
large_receive  no         Enable receive TCP segment aggregation                  True
largesend      1          Enable Hardware Transmit TCP Resegmentation             True
netaddr        9.3.4.1    Address to ping                                          True
pvid           1          PVID to use for the SEA device                          True
pvid_adapter   ent2       Default virtual adapter to use for non-VLAN-tagged packets True
real_adapter   ent0       Physical adapter associated with the SEA                 True
```

thread	1	Thread mode enabled (1) or disabled (0)	True
virt_adapters	ent2	List of virtual adapters associated with the SEA (comma separated)	True

- Define the network configuration for each Virtual I/O Server on the newly created SEA interface. You can use the **mktcpip** command or the **cfgassist** command to launch the SMIT menus, as shown in Figure 16-74 for VIO_Server2.

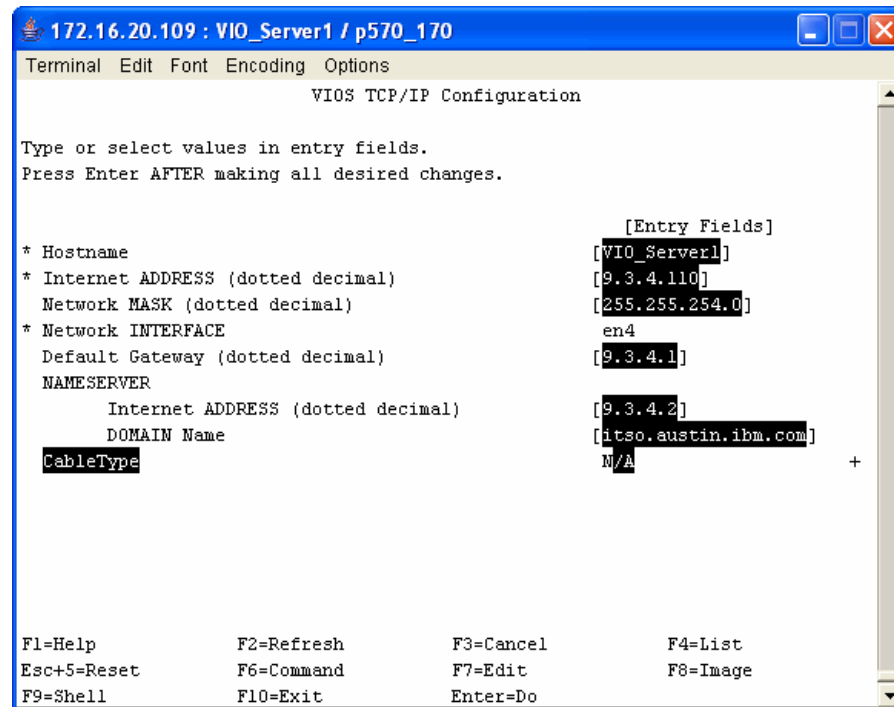


Figure 16-74 Create an IP on the Shared Ethernet Adapter using **cfgassist**

- Create the client partitions following the instructions in "Creating a client partition" on page 354. Each client partition needs to be configured with one virtual Ethernet adapter in VLAN 1.

Testing Shared Ethernet Adapter failover

Perform the following steps to test whether the SEA failover configuration works as expected:

- Open a remote session from a system on the external network to any of the client partitions. If your session gets disconnected during any of the tests your configuration is not highly available. You might want to run a **ping** command to verify that you have continuous connectivity.

2. On the VIO_Server1, check whether the primary adapter is active using the **entstat** command:

```
$ entstat -all ent4 | grep Active
Priority: 1 Active: True
```

3. Perform a manual SEA failover using the following **chdev** command to switch to the standby adapter:

```
$ chdev -dev ent4 -attr ha_mode=standby
```

4. Check whether the SEA failover was successful using the **entstat** command. On VIO_Server1 the entstat output has to look like this:

```
$ entstat -all ent4 | grep Active
Priority: 1 Active: False
```

You must also see the following entry in the errorlog when you issue the **errlog** command:

```
40D97644 1205135007 I H ent4 BECOME BACKUP
```

On VIO_Server2 the entstat output has to look like this:

```
$ entstat -all ent4 | grep Active
Priority: 2 Active: True
```

You must see the following entry in the errorlog when you issue the **errlog** command.

```
E136EAFA 1205135007 I H ent4 BECOME PRIMARY
```

Important: You might experience up to 30 seconds delay in failover when using SEA failover. The behavior depends on the network switch and the spanning tree settings.

5. On VIO_Server1, switch back to the primary adapter and verify that the primary adapter is active using these commands:

```
$ chdev -dev ent4 -attr ha_mode=auto
ent4 changed
$ entstat -all ent4 | grep Active
Priority: 1 Active: True
```
6. Unplug the link of the physical adapter on VIO_Server1. Use the entstat command to check whether the SEA has failed over to the standby adapter.
7. Re-plug the link of the physical adapter on VIO_Server1 and verify that the SEA has switched back to the primary.

Shared Ethernet Adapter failover with Load Sharing

The Virtual I/O Server Version 2.2.1.0, or later, provides a load sharing function to enable to use the bandwidth of the backup Shared Ethernet Adapter (SEA).

The SEA failover configuration provides redundancy by configuring a primary and backup SEA pair on Virtual I/O Servers (VIOS). The backup SEA is in standby mode, and is used when the primary SEA fails. The bandwidth of the backup SEA is not used in normal operation.

Figure 16-75 shows a basic SEA failover configuration. All network packets of all Virtual I/O clients are bridged by the primary VIOS.

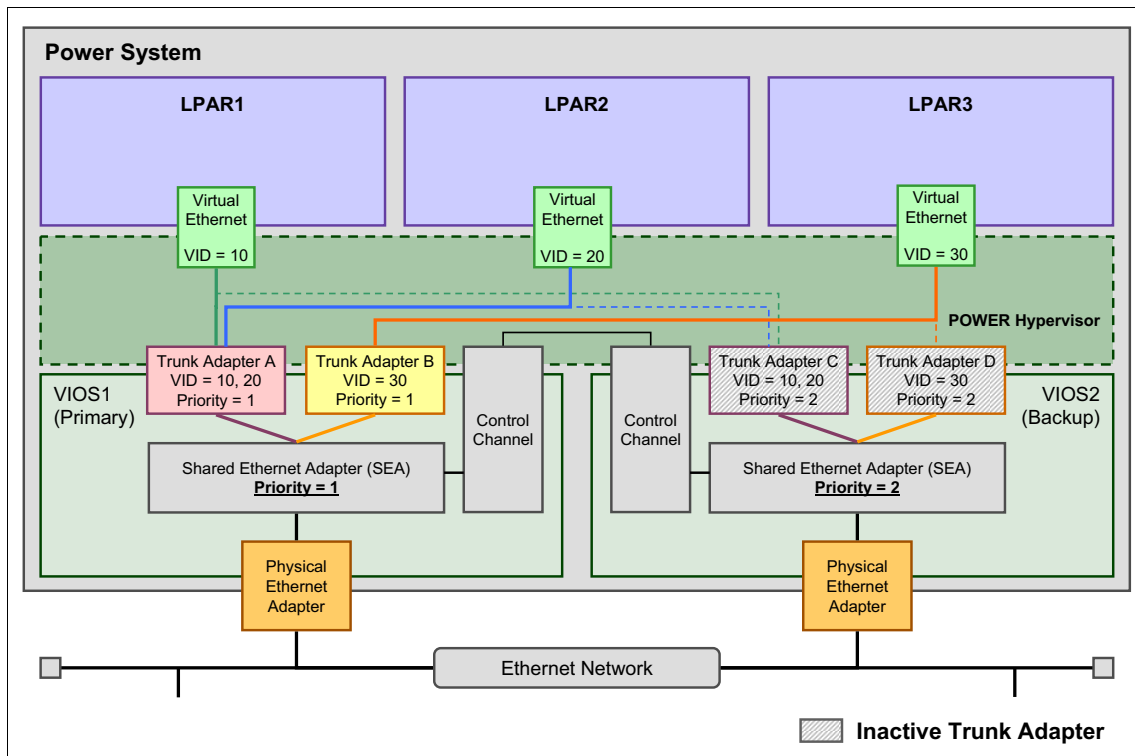


Figure 16-75 SEA failover Primary-Backup configuration

If you need more detailed information for the SEA failover concepts and how to configure SEA failover environment, see “SEA failover” on page 591.

On the other hand, SEA failover with Load Sharing makes effective use of the backup SEA bandwidth, as shown in Figure 16-76. In this example, network packets of LPAR1 and LPAR2 are bridged by VIOS2, and LPAR3 is bridged by VIOS1.

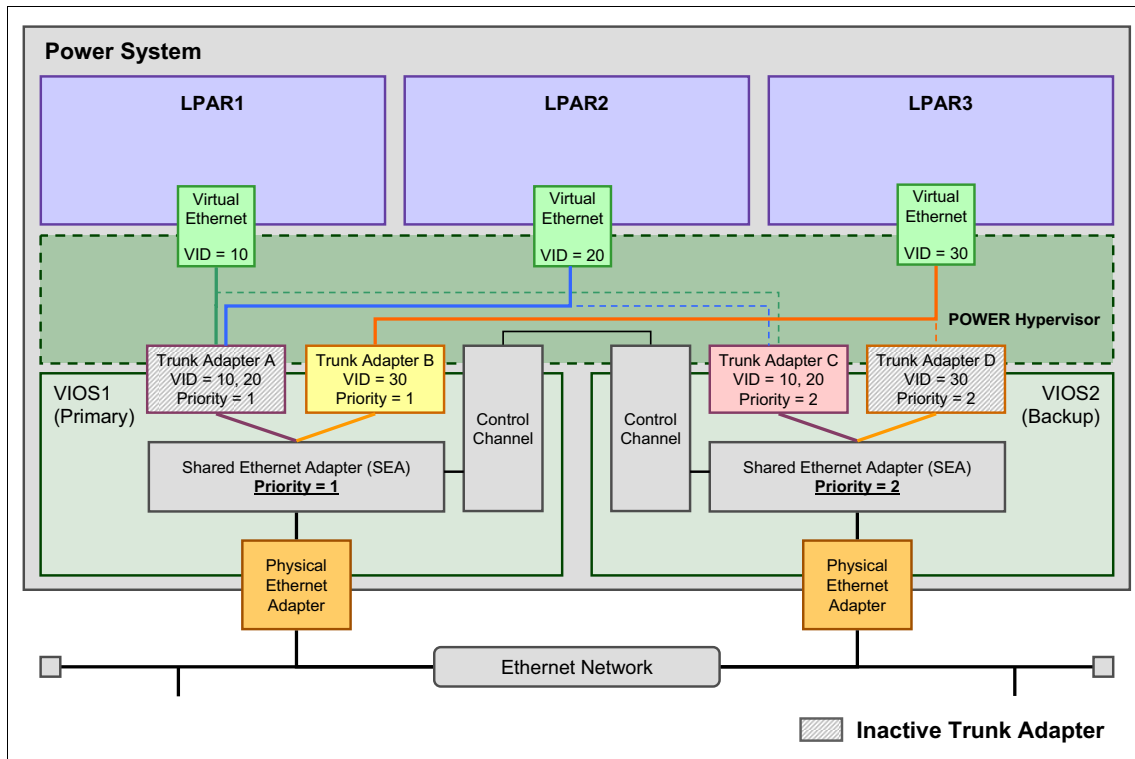


Figure 16-76 SEA failover with Load Sharing

Prerequisites and requirements for SEA failover with Load Sharing are as follows:

- ☐ Both primary and backup Virtual I/O Servers are at Version 2.2.1.0, or later.
- ☐ Two or more trunk adapters are configured for the primary and backup SEA pair.
- ☐ Load Sharing mode must be enabled on both primary and backup SEA pair.
- ☐ The virtual local area network (VLAN) definitions of the trunk adapters are identical between the primary and backup SEA pair.

Important: You need to set the same priority to all trunk adapters under one SEA. The primary and backup priority definitions are set at the SEA level, not at trunk adapters level.

You can check these prerequisites and requirements in this sample, SEA failover with Load Sharing configuration shown as in Figure 16-76 on page 598.

- ▶ Both VIOS1 and VIOS2 should be at Version 2.2.1.0, or later.
- ▶ Two trunk adapters, Adapter A and B, are configured on the primary SEA on VIOS1, and Adapter C and D are configured on the backup SEA on VIOS2.
- ▶ All of the VLAN definitions of trunk adapters match. The primary SEA on VIOS1 has Adapter A with VLANs 10 and 20, and the backup SEA on VIOS2 has Adapter C with VLANs 10 and 20. The Adapter B and D is the same.

Configuring SEA failover with Load Sharing mode is the same as configuring SEA in failover mode. You set **ha_mode** to **sharing** instead of **auto** when you create a SEA (Example 16-93).

Example 16-93 Creating an SEA (ent7) with Load Sharing mode

```
$ mkvdev -sea ent1 -vadapter ent4,ent5 -default ent4 -defaultid 10 -attr ha_mode=sharing
ctl_chan=ent6
ent7 Available
```



```
$ lsdev -dev ent7 -attr
```

attribute	value	description	user_settable
accounting	disabled	Enable per-client accounting of network statistics	True
ctl_chan	ent6	Control Channel adapter for SEA failover	True
gvrp	no	Enable GARP VLAN Registration Protocol (GVRP)	True
ha_mode	sharing	High Availability Mode	True
jumbo_frames	no	Enable Gigabit Ethernet Jumbo Frames	True
large_receive	no	Enable receive TCP segment aggregation	True
largesend	1	Enable Hardware Transmit TCP Resegmentation	True
lldpsvc	no	Enable IEEE 802.1qbg services	True
netaddr	0	Address to ping	True
pvid	10	PVID to use for the SEA device	True
pvid_adapter	ent4	Default virtual adapter to use for non-VLAN-tagged packets	True
qos_mode	disabled	N/A	True
real_adapter	ent1	Physical adapter associated with the SEA	
True			
thread	1	Thread mode enabled (1) or disabled (0)	True
virt_adapters	ent4,ent5	List of virtual adapters associated with the SEA (comma separated)	True

You have already configured an SEA failover environment in failover mode; you can change the **ha_mode** attribute from **auto** to **sharing** by using the **chdev** command dynamically. You can also add a new trunk adapter to the existing SEA if you need, as shown in Example 16-94.

Example 16-94 Adding a trunk adapter and changing SEA (ent6) failover mode

```
Adding a trunk adapter to the SEA.
    ent6: SEA
    ent4: trunk adapter which is already part of the SEA
    ent7: new trunk adapter adding to the SEA
$ chdev -dev ent6 -attr virt_adapters=ent4,ent7
ent6 changed

Changing the SEA to Load Sharing mode.
$ chdev -dev ent6 -attr ha_mode=sharing
ent6 changed

$ lsdev -dev ent6 -attr
attribute      value      description                                     user_settable
accounting     disabled  Enable per-client accounting of network statistics      True
ctl_chan       ent5      Control Channel adapter for SEA failover                 True
gvrp           no        Enable GARP VLAN Registration Protocol (GVRP)           True
ha_mode       sharing  High Availability Mode                                   True
jumbo_frames   no        Enable Gigabit Ethernet Jumbo Frames                    True
large_receive  no        Enable receive TCP segment aggregation                  True
largesend      1         Enable Hardware Transmit TCP Resegmentation             True
lldpsvc        no        Enable IEEE 802.1qbg services                           True
netaddr        0         Address to ping                                          True
pvid           10       PVID to use for the SEA device                          True
pvid_adapter   ent4      Default virtual adapter to use for non-VLAN-tagged packets True
qos_mode       disabled  N/A                                                       True
real_adapter   ent1      Physical adapter associated with the SEA                 True
thread         1         Thread mode enabled (1) or disabled (0)                 True
virt_adapters ent4,ent7 List of virtual adapters associated with the SEA (comma separated) True
```

If you need to disable it while it is running in load sharing mode, set the **ha_mode** to any value other than **sharing**, for example **standby**, **auto**, or **disable**.

Important: To create or enable the SEA failover with Load Sharing, you have to enable the load sharing mode on the primary SEA first before enabling load sharing mode on the backup SEA.

To change the **ha_mode** from **sharing** to **auto**, disable the load sharing mode, and set **ha_mode** to **auto** on the primary SEA first. Then set it on the backup to minimize the chance of a broadcast storm of the SEA.

The **entstat** command provides detailed information for the current SEA status such as State, Trunk Adapter Priority, and VLAN IDs. The output of **entstat** consists of some statistics for physical and virtual adapters in the SEA, as shown in Example 16-95.

Example 16-95 Statistics for adapters in the Shared Ethernet Adapter

```
$ entstat -all ent6
-----
ETHERNET STATISTICS (ent6) :
Device Type: Shared Ethernet Adapter
Hardware Address: 00:1a:64:bb:69:49
Elapsed Time: 0 days 10 hours 44 minutes 30 seconds

Transmit Statistics:                Receive Statistics:
-----
...
-----
Statistics for adapters in the Shared Ethernet Adapter ent6
-----
...
VLAN Ids :
    ent4: 10
    ent7: 30 130
Real Side Statistics:
    Packets received: 34275
...
Type of Packets Received:
    ...
    Limbo Packets: 0
    State: PRIMARY_SH
    Bridge Mode: Partial
    VID shared: 10
    Number of Times Server became Backup: 0
    Number of Times Server became Primary: 1
    High Availability Mode: Sharing
    Priority: 1
-----
Real Adapter: ent1

ETHERNET STATISTICS (ent1) :
Device Type: 4-Port 10/100/1000 Base-TX PCI-X Adapter (14101103)
...
-----
Virtual Adapter: ent4

ETHERNET STATISTICS (ent4) :
Device Type: Virtual I/O Ethernet Adapter (1-lan)
...
```

Virtual I/O Ethernet Adapter (l-lan) Specific Statistics:

RQ Length: 4481
Trunk Adapter: True
 Priority: 1 Active: True
Filter MCast Mode: False
...
Port VLAN ID: 10
VLAN Tag IDs: None
...

Virtual Adapter: ent7

ETHERNET STATISTICS (ent7) :
Device Type: Virtual I/O Ethernet Adapter (l-lan)
...

Virtual I/O Ethernet Adapter (l-lan) Specific Statistics:

RQ Length: 4481
Trunk Adapter: True
 Priority: 1 Active: False
Filter MCast Mode: False
...
Port VLAN ID: 130
VLAN Tag IDs: 30
...

Control Channel Adapter: ent5
...

This example shows that the SEA consists of one physical adapter, two trunk adapters with VLAN ID 10 and 30, and one control channel adapter. The details are as follows:

- ▶ State: PRIMARY_SH
The “_SH” means that the SEA is running in load sharing mode. You can also see the status as High Availability Mode: Sharing.
- ▶ Priority:1 Active: True
This shows that the trunk adapter is configured as a part of the primary SEA and the adapter is activated.
- ▶ Priority:1 Active: False
This shows that the trunk adapter is configured as a part of the primary SEA and the adapter is deactivated.

Tip: The load sharing algorithm automatically determines which trunk adapters will be activated and will treat network packets for VLANs in the SEA pair. You cannot specify the active trunk adapters of the SEAs manually in the load sharing mode.

16.3.3 EtherChannel Backup in the AIX client

EtherChannel Backup (ECB) can be used on AIX client partitions as a solution of Network Interface Backup (NIB) to provide redundant access to external networks when two Virtual I/O Servers are used. Figure 16-77 shows a sample configuration. To keep the picture simple, only the AIX client partition DB_Server is shown.

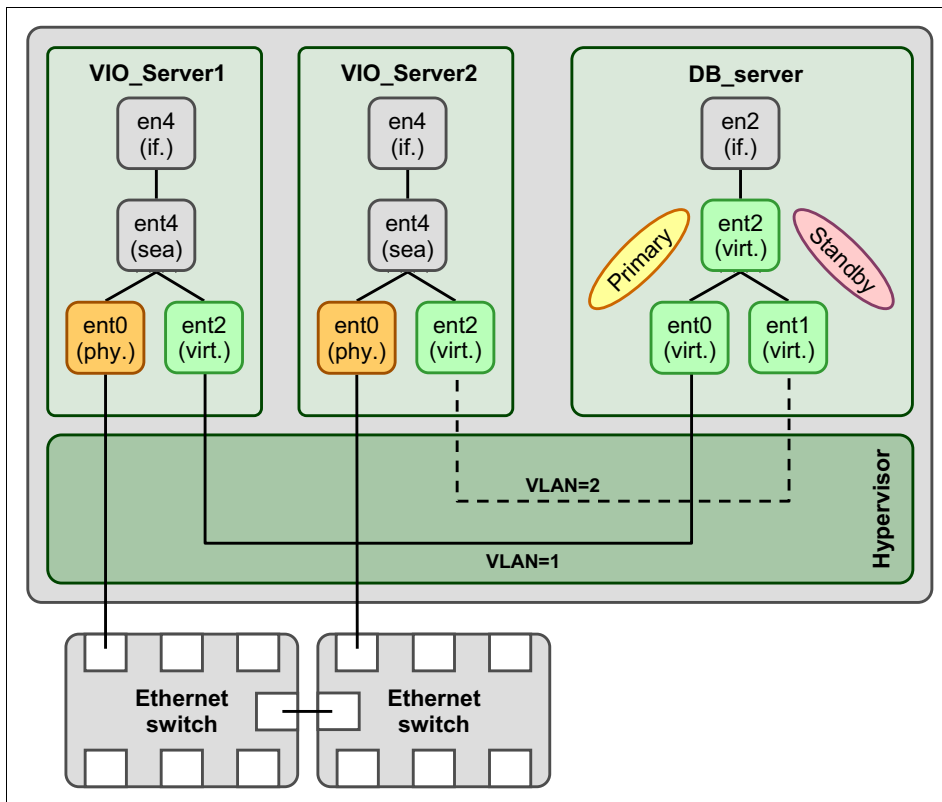


Figure 16-77 ECB configuration on AIX client

In a ECB configuration, the Shared Ethernet Adapters (SEA) on the two Virtual I/O Servers are each configured on a different VLAN. In our example, VIO_Server1 is configured on VLAN 1 and VIO_Server2 is configured on VLAN 2. Each Virtual I/O Server therefore needs a virtual Ethernet Adapter configured with the correct VLAN ID.

The client partition has two virtual Ethernet Adapters. One of them is configured on VLAN 1 providing the connection to the SEA in VIO_Server1. The second one is configured on VLAN 2 connecting to VIO_Server2. These two adapters compose an EtherChannel Backup configuration.

Configuring EtherChannel Backup

Use the following steps to set up the scenario:

1. Create two Virtual I/O Server partitions and name them VIO_Server1 and VIO_Server2, following the instructions in 12.1, “Creating a Virtual I/O Server” on page 312. In step 10, select one Ethernet adapter and one storage adapter.
2. Install both Virtual I/O Servers by following the instructions in 12.2, “Installation of Virtual I/O Server” on page 333.
3. Configure each Virtual I/O Server with one virtual Ethernet adapter. Each Virtual I/O Server needs to be on a different VLAN. See Table 16-9.

Table 16-9 EtherChannel Backup configuration examples

Virtual I/O Server	Virtual I/O Server slot	VLAN
VIO_Server1	10	1
VIO_Server2	10	2

4. On each Virtual I/O Server, define a Shared Ethernet Adapter with the correct VLAN ID. Example 16-96 shows the configuration of the SEA on VIO_Server1 with VLAN ID 1 through the *defaultid* flag. Example 16-97 shows the configuration of the SEA on VIO_Server2 with VLAN ID 2.

Example 16-96 SEA adapter configuration on VIO_Server1

```
$ mkvdev -sea ent0 -vadapter ent2 -default ent2 -defaultid 1
ent4 Available
en4
et4
```

Example 16-97 SEA adapter configuration on VIO_Server2

```
$ mkvdev -sea ent0 -vadapter ent2 -default ent2 -defaultid 2
ent4 Available
en4
et4
```

5. Also add virtual SCSI adapters to have a highly available virtual SCSI configuration as described in “Availability configurations using multipathing” on page 501 or “Availability configurations using mirroring” on page 534
6. Create the client partitions following the instructions in “Creating a client partition” on page 354. Each client partition needs to be configured with one virtual Ethernet adapter in VLAN1 and one in VLAN2.
7. In each client partition, define the EtherChannel. Enter **smi**t and select **Devices** → **Communication** → **EtherChannel / IEEE 802.3ad Link Aggregation** → **Add An EtherChannel / Link Aggregation** or using the **smitty etherchannel** command. Select the primary adapter, ent0, and press Enter. Example 16-98 shows the SMIT menu for adding an EtherChannel/Link Aggregation.

Example 16-98 Add An EtherChannel / Link Aggregation smit menu

Add An EtherChannel / Link Aggregation

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

	[Entry Fields]	
EtherChannel / Link Aggregation Adapters	ent0	+
Enable Alternate Address	no	+
Alternate Address	[]	+
Enable Gigabit Ethernet Jumbo Frames	no	+
Mode	standard	+
Hash Mode	default	+
Backup Adapter	ent1	+
Automatically Recover to Main Channel	yes	+
Perform Lossless Failover After Ping Failure	yes	+
Internet Address to Ping	[9.3.4.1]	
Number of Retries	[]	+#
Retry Timeout (sec)	[]	+#

Esc+1=Help	Esc+2=Refresh	Esc+3=Cancel	Esc+4=List
Esc+5=Reset	F6=Command	F7=Edit	F8=Image
F9=Shell	F10=Exit	Enter=Do	

Tip: The following message, which appears in the errorlog when the EtherChannel is created, can be ignored:

```
5561971C 1123194707 P S ent2 UNSUPPORTED IOCTL IN DEVICE DRIVER
```

Testing EtherChannel Backup

Perform the following steps to verify that the ECB configuration works as expected. The procedure shows how failover works when the network connection of a Virtual I/O Server is disconnected. The failover works in the same fashion when a Virtual I/O Server is rebooted. You can test this by rebooting VIO_Server1 instead of unplugging the network cable in step 2.

Follow these steps to verify the ECB configuration:

1. Do a remote login to the client partition using Telnet or SSH. After you are logged in, check whether the primary channel that is connected to VIO_Server1 is active using the **entstat** command, as shown in Example 16-99.

Example 16-99 Verifying the active channel in an EtherChannel

```
# entstat -d ent2 | grep Active  
Active channel: primary channel
```

2. Unplug the network cable from the physical network adapter that is connected to VIO_Server1.

3. As soon as the EtherChannel notices that it has lost connection, it has to perform a switchover to the backup adapter. You will see a message as shown in Example 16-100 in the errorlog.

Important: Your telnet or SSH connection must not be disconnected. If it is disconnected, your configuration is not highly available.

Example 16-100 Errorlog message when primary channel fails

LABEL: ECH_PING_FAIL_PRMRY
IDENTIFIER: 9F7B0FA6

Date/Time: Fri Nov 23 19:53:35 CST 2007
Sequence Number: 141
Machine Id: 00C1F1704C00
Node Id: NIM_server
Class: H
Type: INFO
WPAR: Global
Resource Name: ent2
Resource Class: adapter
Resource Type: ibm_ech
Location:

Description
PING TO REMOTE HOST FAILED

Probable Causes
CABLE
SWITCH
ADAPTER

Failure Causes
CABLES AND CONNECTIONS

Recommended Actions
CHECK CABLE AND ITS CONNECTIONS
IF ERROR PERSISTS, REPLACE ADAPTER CARD.

Detail Data
FAILING ADAPTER
PRIMARY
SWITCHING TO ADAPTER

```
ent1
Unable to reach remote host through primary adapter: switching over
to backup adapter
```

As shown in Example 16-101, the **entstat** command will also show that the backup adapter is now active.

Example 16-101 Verifying the active channel in an EtherChannel

```
# entstat -d ent2 | grep Active
Active channel: backup adapter
```

4. Reconnect the physical adapter in VIO_Server1.
5. The EtherChannel will not automatically switch back to the primary channel. In the SMIT menu, there is an option to “Automatically Recover to Main Channel”. It is set to Yes by default and this is the behavior when using physical adapters. However, virtual adapters do not adhere to this. Instead, the backup channel is used until it fails and then switches to the primary channel. You have to manually switch back using the **/usr/lib/methods/ethchan_config -f** command as shown in Example 16-102. A message will appear in the errorlog when the EtherChannel recovers to the primary channel.

Example 16-102 Manual switch to primary channel using entstat

```
# /usr/lib/methods/ethchan_config -f ent2
# entstat -d ent2 | grep Active
Active channel: primary channel
# errpt
8650BE3F 1123195807 I H ent2 ETHERCHANNEL RECOVERY
```

Tip: You can use the **dsh** command to automate the check or switch back if you have a lot of client partitions.

16.3.4 IBM i virtual IP address failover for virtual Ethernet adapters

An equivalent solution to Network Interface Backup can be implemented for IBM i using a virtual IP address (VIPA) with a manually specified list of preferred interfaces for proxy Address Resolution Protocol (ARP) agent selection and a virtual-to-virtual failover script.

The VIPA configuration of the IBM i client partition with two virtual Ethernet adapters each connected to a redundant Virtual I/O Server partition acting as the layer-2 bridge to the external network is shown in Figure 16-78.

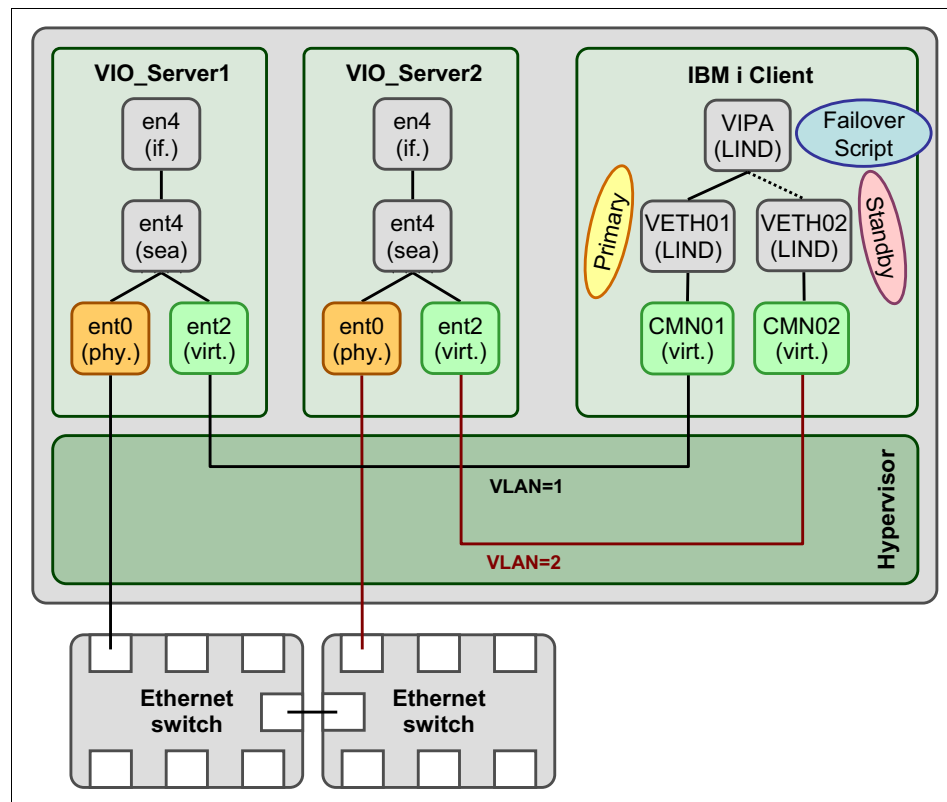


Figure 16-78 VIPA failover configuration for IBM i client

Unlike other IP interfaces a VIPA is not bound to a specific line description (LIND). Instead the user can specify a *preferred list* of prioritized existing IP interfaces that serve as proxies to connect the VIPA via the first active interface in the list to the network. An example of a configuration of two IP interfaces for the virtual Ethernet adapters with line descriptions VETH01 and VETH02 and one corresponding VIPA interface is shown in Example 16-103.

Example 16-103 IBM i interface configuration for VIPA

```
ADDTCPIFC INTNETADR('192.168.1.2') LIND(VETH01) SUBNETMASK('255.255.255.0')
ADDTCPIFC INTNETADR('192.168.1.3') LIND(VETH02) SUBNETMASK('255.255.255.0')
ADDTCPIFC INTNETADR('192.168.1.4') LIND(*VIRTUALIP)
SUBNETMASK('255.255.255.255') PREFIFC('192.168.1.2' '192.168.1.3')
```

For further information about IBM i VIPA configurations, see the *IBM i Information Center* at this website:

<http://publib.boulder.ibm.com/infocenter/iserics/v7r1m0/index.jsp?topic=%2Frzajw%2Frzajw1bvip.htm>

Since physical Ethernet link changes are not propagated to the virtual Ethernet adapters of the IBM i client a scripting solution is required to implement a VIPA failover based on periodic health checks for each virtual Ethernet based interface in the VIPA's preferred list.

The IBM i CL program shown in Example 16-104 performs a periodic health check about every 30 seconds for each of the two interfaces 192.168.1.2 and 192.168.1.3 on the VIPA's preferred list by monitoring ping responses for a remote target IP address 192.168.1.1. It enables automatic VIPA failover for virtual Ethernet adapters by rearranging the interfaces on the VIPA's preferred list if the first interface becomes unavailable while the second is available.

Example 16-104 IBM i CL program for VIPA virtual-to-virtual failover

```
PGM
DCL &TARGET *CHAR 15 '192.168.1.1'
DCL &IFC1 *CHAR 15 '192.168.1.2'
DCL &IFC2 *CHAR 15 '192.168.1.3'
DCL &VIPA *CHAR 15 '192.168.1.4'

DCL &IFC1STAT *INT 4 0
DCL &IFC2STAT *INT 4 0
DCL &PREF *INT 4 1

/* MAKE SURE PROXY IN PREFERRED ORDER */
CHGTCPIFC &VIPA PREFIFC(&IFC1 &IFC2)

/* BEGIN LOOP FOREVER TO MONITOR THE INTERFACES */
LOOP:
/* PING THROUGH INTERFACE 1 AND COUNT FAILURES */
PING &TARGET LCLINTNETA(&IFC1) NBRPKT(1) MSGMODE(*VERBOSE *ESCAPE)
MONMSG MSGID(TCP3210) EXEC(GOTO IFC1FAIL)
CHGVAR &IFC1STAT 0
GOTO IFC2
IFC1FAIL:
IF (&IFC1STAT < 9999) +
THEN(CHGVAR &IFC1STAT (&IFC1STAT + 1))

IFC2:
/* PING THROUGH INTERFACE 2 AND COUNT FAILURES */
```

```

PING &TARGET LCLINTNETA(&IFC2) NBRPKT(1) MSGMODE(*VERBOSE *ESCAPE)
MONMSG MSGID(TCP3210) EXEC(GOTO IFC2FAIL)
CHGVAR &IFC2STAT 0
GOTO UPDVIPA
IFC2FAIL:
  IF (&IFC2STAT < 9999) +
  THEN(CHGVAR &IFC2STAT (&IFC2STAT + 1))

UPDVIPA:
  /* TREAT 2 CONSECUTIVE FAILURES AS LINK DOWN INDICATION */
  IF (&IFC1STAT < 2) THEN(GOTO PREF1)
  IF (&IFC2STAT < 2) THEN(GOTO PREF2)
  /* NO INTERFACES APPEAR TO BE UP... NO CHANGES */
  GOTO DELAY

PREF1:
  IF (&PREF ^= 1) +
  THEN(CHGTCPIFC &VIP A PREFIFC(&IFC1 &IFC2))
  CHGVAR &PREF 1
  GOTO DELAY
PREF2:
  IF (&PREF ^= 2) +
  THEN(CHGTCPIFC &VIP A PREFIFC(&IFC2 &IFC1))
  CHGVAR &PREF 2
  GOTO DELAY

DELAY:
  DLYJOB 30
  GOTO LOOP
ENDPGM

```

16.3.5 Linux Ethernet connection bonding

Ethernet connection bonding in a Linux on POWER client partition can be used to achieve network redundancy when using two Virtual I/O Servers.

Overview

In such a configuration each Virtual I/O Server provides an active Ethernet connection which is combined in a bonding device. For more details on the bonding configuration, see this website:

<http://www.ibm.com/developerworks/wiki/display/LinuxP/Bonding+configuration>

Because there is no hardware link failure for virtual Ethernet devices to trigger a failover to the other network adapter interface, backup mode with ARP broadcast is used. The kernel must support the `arp_validation` feature. This is the case starting with kernel version 2.6.19.

The bonding options are set during the module load. They are added to the `/etc/modprobe.conf` as shown in Example 16-105.

Example 16-105 /etc/modprobe.conf example

```
alias bond0 bonding
options bond0 mode=active-backup arp_interval=1000
arp_ip_target=9.156.175.1,9.156.175.8 primary=eth0 arp_validate=all
alias eth0 ibmveth
alias eth1 ibmveth
alias scsi_hostadapter ibmvscsic
```

After the module is loaded, the interfaces have to be configured. Example 16-106 shows an example configuration in the `/etc/sysconfig/network-scripts/ifcfg-bond0` file.

Example 16-106 /etc/sysconfig/network-scripts/ifcfg-bond0

```
DEVICE=bond0
BOOTPROTO=static
BROADCAST=9.3.5.255
IPADDR=9.3.5.115
NETMASK=255.255.255.0
NETWORK=9.3.5.0
ONBOOT=yes
GATEWAY=9.3.5.1
```

Example 16-107 and Example 16-108 show the configuration files for the slave devices. Make sure that the slave devices have no IP addresses defined and that there are no definitions left from previous configurations. The **ifconfig -a** command only shows the IP address assigned to all the devices.

Example 16-107 etc/sysconfig/network-scripts/ifcfg-eth0

```
DEVICE=eth0
BOOTPROTO=none
ONBOOT=yes
MASTER=bond0
SLAVE=yes
USERCTL=no
```

Example 16-108 etc/sysconfig/network-scripts/ifcfg-eth1

```
DEVICE=eth1
BOOTPROTO=none
ONBOOT=yes
MASTER=bond0
SLAVE=yes
USERCTL=no
```

Tip: In most installation scenarios, SEA failover is easier to use and to maintain and therefore is the best solution.

Testing Ethernet connection bonding

To verify that the device mapper multipath configuration works as expected, perform the following steps:

1. Verify that both links are up as shown here. The status of both slave interfaces must be up, and the link failure count must be 0. In the following example, the active slave adapter is eth0.

```
# cat /proc/net/bonding/bond0
Ethernet Channel Bonding Driver: v2.6.3-rh (June 8, 2005)
```

```
Bonding Mode: fault-tolerance (active-backup)
Primary Slave: None
Currently Active Slave: eth0
MII Status: up
MII Polling Interval (ms): 0
Up Delay (ms): 0
Down Delay (ms): 0
```

```
Slave Interface: eth0
```

MII Status: up
Link Failure Count: 0
Permanent HW addr: ba:d3:f0:00:40:02

Slave Interface: eth1
MII Status: up
Link Failure Count: 0
Permanent HW addr: ba:d3:f0:00:40:03

2. Unplug the connection of the SEA in one of the Virtual I/O Servers to the external network. In our example we are unplugging the link in VIO_Server1 to which the slave interface eth0 is connected. As you can see in the example here, the slave interface is going to status down. The currently active slave interface has changed to eth1.

```
# cat /proc/net/bonding/bond0  
Ethernet Channel Bonding Driver: v2.6.3-rh (June 8, 2005)
```

Bonding Mode: fault-tolerance (active-backup)
Primary Slave: None
Currently Active Slave: eth1
MII Status: up
MII Polling Interval (ms): 0
Up Delay (ms): 0
Down Delay (ms): 0

Slave Interface: eth0
MII Status: down
Link Failure Count: 1
Permanent HW addr: ba:d3:f0:00:40:02

Slave Interface: eth1
MII Status: up
Link Failure Count: 0
Permanent HW addr: ba:d3:f0:00:40:03

3. You can also see the status of the Ethernet bonding connection using the **dmesg** command as shown here:

```
bonding: bond0: link status down for active interface eth0,  
disabling it  
bonding: bond0: making interface eth1 the new active one.
```

Tip: After the primary slave interface becomes active again, there is no automatic switch back.

16.3.6 General rules for setting modes for QoS

The following general rules apply to setting modes for QoS.

Use strict mode for the following conditions:

- ▶ When maintaining priority is more important than preventing “starvation”.
- ▶ When the network administrator has a thorough understanding of the network traffic.
- ▶ When the network administrator understands the possibility of overhead and bandwidth starvation, and knows how to prevent this from occurring.

Use loose mode for the following condition:

- ▶ When preventing starvation is more important than maintaining priority.

16.3.7 Denial of Service hardening

A Denial of Service (DoS) attack targets a machine and makes it unavailable. The target machine is bombarded with fake network communication requests for a service (such as ftp, telnet, and so on) that cause it to allocate resources for each request. For every request, a port is held busy and process resources are allocated. Eventually, the target machine is exhausted of all its resources and becomes unresponsive.

Like other operating systems, VIOS/AIX was vulnerable to DoS attacks. To corporations, network security is paramount and it was unacceptable to have servers bogged down by DoS attacks.

Tip: Users can set DoS hardening rules on default ports by using the **viosecure -level high** command. See “Security hardening rules” on page 431 for more details.

Solution

One solution, adopted from IBM z/OS®, is to limit the total number of active connections an application has at one time. This puts a restriction on the number of address spaces created by forking applications such as ftpd, telnetd, and so on. A fair share algorithm is also provided, based on the percentage of remaining available connections already held by a source IP address. A fair share algorithm will enforce TCP traffic regulations policies.

To utilize network traffic regulation, you need to enable it first. Example 16-109 shows how to enable network traffic regulation.

Example 16-109 Enabling network traffic regulation

```
# no -p -o tcptr_enable=1
# no -a |grep tcptr
      tcptr_enable = 1
```

The **tcptr** command can be used to display the current policy for various services, and modify it. For the Virtual I/O Server, you need to execute it from the root shell. It has the following syntax:

```
tcptr -add    <start_port> <end_port> <max> <div>
tcptr -delete <start_port> <end_port>
tcptr -show
```

Where:

<start_port>	This is the starting TCP port for this policy.
<end_port>	This is the ending TCP port for this policy.
<max>	This is the maximum pool of connections for this policy.
<div>	This is the divisor (<32) governing the available pool.

Example 16-110 shows how to regulate network traffic for port 25 (sendmail service).

Example 16-110 Using tcptr for network traffic regulation for sendmail service

```
# tcptr -show
policy: Error failed to allocate memory

(1) root @ core13: 6.1.2.0 (0841A_61D) : /
# tcptr -add 25 25 1000
StartPort=25    EndPort=25    MaxPool=1000    Div=0

(0) root @ core13: 6.1.2.0 (0841A_61D) : /
# tcptr -show
TCP Traffic Regulation Policies:
StartPort=25    EndPort=25    MaxPool=1000    Div=0    Used=0
```



Server virtualization setup

This chapter describes how to set up the server virtualization features such as Live Partition Mobility and partition Suspend and Resume.

It covers the following topics:

- ▶ Live Partition Mobility setup
- ▶ Suspend and Resume setup

17.1 Live Partition Mobility setup

Based on all Live Partition Mobility concepts and considerations presented on the previous chapters, you should be able to proceed and do a system migration.

A functional Live Partition Mobility environment setup can easily be achieved if all the planning requirements were correctly followed and are in place. To help you achieve this, this section will cover a setup checklist and the corresponding configurations to facilitate the enablement of this feature for a migrating a partition.

17.1.1 Live Partition Mobility enablement

To achieve the LPM enablement a checklist follows summarizing the main requirements discussed in 11.2, “Live Partition Mobility planning” on page 260.

Architecture

- ☐ Source and destination systems must be POWER6-based systems or later (POWER7-based systems for Live Partition Mobility with IBM i)

HMC and system firmware levels

- ☐ HMC version - Recommended V7R7.x, where “x” is the latest level allowed according the compatibility matrix available on the site.

<https://www14.software.ibm.com/webapp/set2/flrt/home>

HMC V7R7.5 or later required for LPM with IBM i

- ☐ Redundant HMC - Proper communication is available between both HMC. On the same network or correct Secure Shell enabled route.
- ☐ Redundant HMC to Remote Migration - Authentication keys between both HMC hosting source and destination servers set. Redundant HMCs should be at the same level.

Licensing

- ☐ PowerVM Enterprise Edition license: Both source and destination systems must have the PowerVM Enterprise Edition license code installed. Or if you want to try LPM there is a Trial LPM License you can obtain that will enable LPM for 60 days.

Network

- ☐ Virtual network adapter - Ensure that the mobile LPARs do not have physical network adapters.
- ☐ Shared Ethernet Adapter SEA - Must to be created on the target and destination VIOS
- ☐ VLAN - If using VLAN tag, ensure that they are properly configured and working on all VIOS.
- ☐ RMC connection - Ensure a working RMC connection for the Virtual I/O Servers on the source and destination system. For active migration a working RMC connection is also required for the mobile partition.

Storage

- ☐ LUN - The LUNs used for virtual SCSI must be mapped to the Virtual I/O Servers on both systems which requires also that the SAN fabric zoning to be implemented such that the Virtual I/O Servers on both systems have access to same backing physical storage devices.
 - If using NPIV ensure that both virtual WWPN addresses of the virtual Fibre Channel client adapters are correctly zoned and configured in the SAN storage environment.
- ☐ SCSI Reservation - SCSI reservation must be disabled.

Virtual I/O Server

- ☐ Virtual I/O Server Level - It is recommended to be at the latest level to obtain the best results.
- ☐ **Mover service partition** - Ensure that for *active* partition migration at least one of the mover service partitions (MSP) is enabled on a source and destination Virtual I/O Server partition.

Source and Destination System Setup

- ☐ Battery power - Ensure that the destination system is not running on battery power.
- ☐ Ensure that the destination system has enough available memory to support the mobile partition.
- ☐ Logical memory block size (LMB)- Ensure that the logical memory block (LMB) size is the same on the source and destination systems. This can be checked using the Advanced System Management Interface (ASMI).
- ☐ Ensure that the destination system has enough available processors (or processing units) to support the mobile partition.

- ☐ If the mobile partition uses dedicated processors, the destination system must have at least this number of available processors.
- ☐ If the mobile partition is assigned to a Shared-Processor Pool, the destination system must have enough spare entitlement to allocate it to the mobile partition in the destination Shared-Processor Pool.
- ☐ The destination system have a virtual switch configured with the same name as the source system.

Active migrations setup

You can perform an *active* partition migration if the following requirements are met. If this is not the case, you can still run an inactive partition migration:

- ☐ The partition is in the Running state.
- ☐ The partition does not use huge pages. Check this in the advanced properties of the partition on the HMC.
- ☐ The partition does not use the Barrier Synchronization Register. Check that the “number of BSR arrays” is set to zero (0) in the memory properties of the partition on the HMC. Changing this setting currently requires a reboot of the partition. The partition must not contain any Physical I/O.

Mobile partition for mobility setup

The requirements below needed to be accomplish independently of the type of the migration that will be performed:

- ☐ RMC connections - For active partition migration, ensure that Resource Monitoring and Control (RMC) connections are established. Note, IBM i partitions do not use the HMC.
- ☐ Disable redundant error path reporting - Ensure that the mobile partition is not enabled for redundant error path reporting.
- ☐ Virtual serial adapters - Ensure that the mobile partition is not using a virtual serial adapter in slots higher than slot 1.
- ☐ Partition workload groups - Ensure that the mobile partition is not part of a logical partition group.
- ☐ Barrier-synchronization register - Ensure that the mobile partition is not using barrier-synchronization register (BSR) arrays.
- ☐ Huge pages - Ensure that the mobile partition is not using huge pages.
- ☐ An IBM i mobile partition must have “Restricted IO Partition” enabled in its partition profile

17.1.2 Live Partition Mobility setup

In this section the required configuration setup to allow for Live Partition Mobility operations is described.

Remote HMC communication

Perform the following configuration steps if using a remote migration, that is, migrating a partition to a destination system managed by another *remote* HMC.

Test network communication

To test the network communication between the two HMC systems involved in the migration, proceed as follows:

1. In the navigation area, select **HMC Management**.
2. In the Operations section of the contents area, select **Test Network Connectivity**.
3. In the Network Diagnostic Information window, select the **Ping** tab.
4. In the text box, enter the IP address or host name of the remote HMC, and click **Ping**.
5. Review the results to ensure that certain packets were not lost, as shown in Figure 17-1:

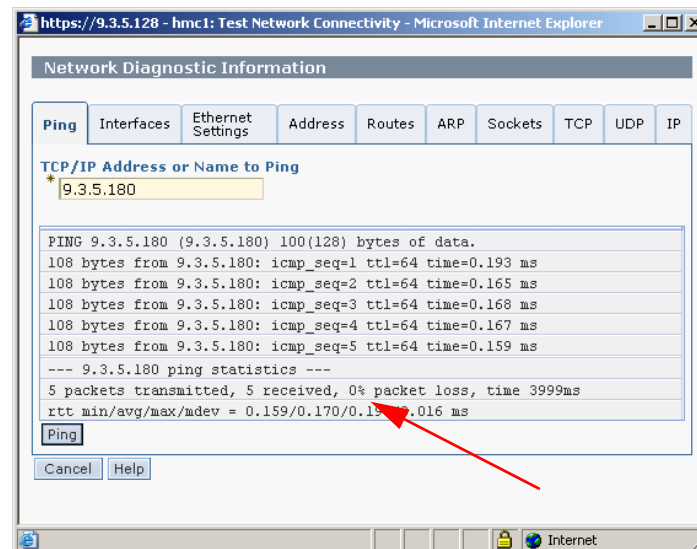


Figure 17-1 Network ping successful to remote HMC

SSH authentication keys

Allow communication between the local and remote HMC through SSH key authentication. The local user must retrieve authentication keys from the user on the remote HMC. This retrieval requires access to the CLI on the local HMC. To allow the local HMC to communicate with remote HMC, first ensure that remote command execution is enabled on the remote HMC.

To enable remote command execution (see Figure 17-2):

1. In the navigation area, select **HMC Management**.
2. In the Administration section of the contents area, select **Remote Command Execution**.

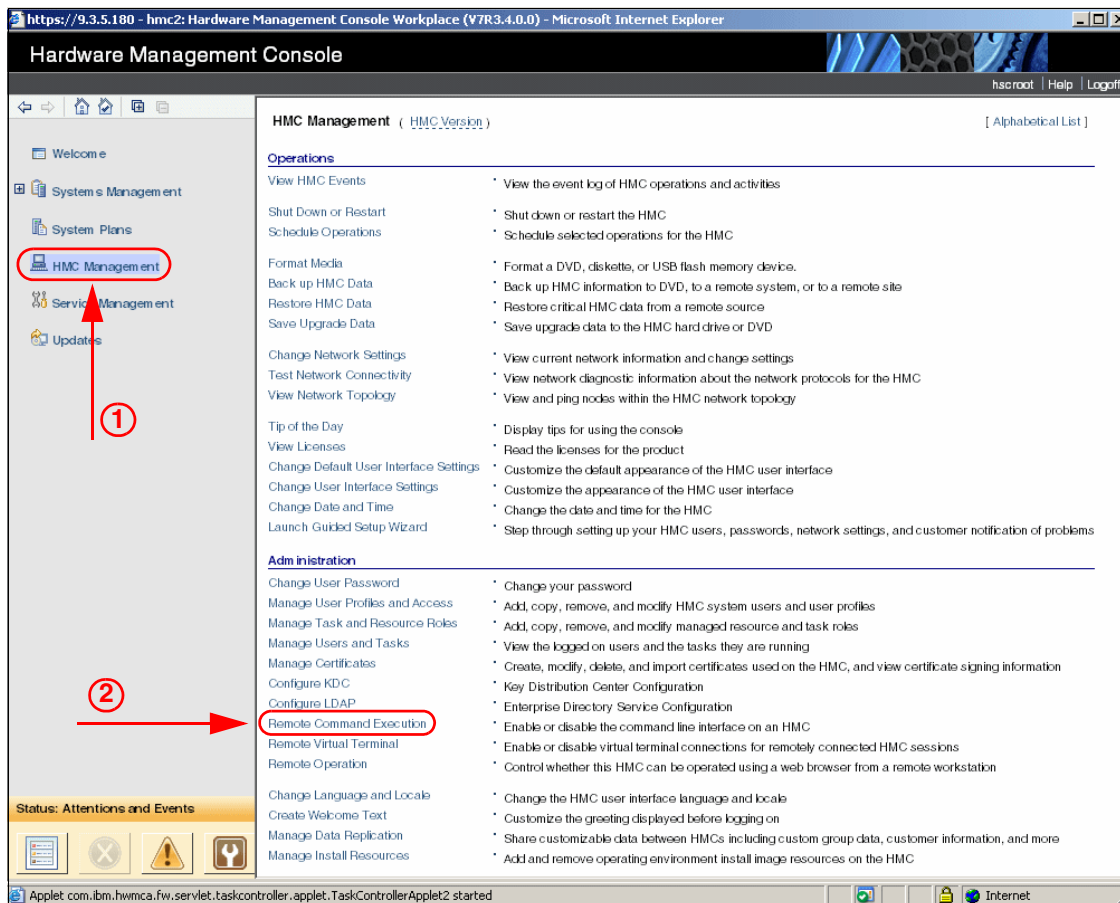


Figure 17-2 HMC option for remote command execution

3. In the Remote Command Execution window, enable the check box to **Enable remote command execution using the ssh facility**, as shown in Figure 17-3. Click **OK**.

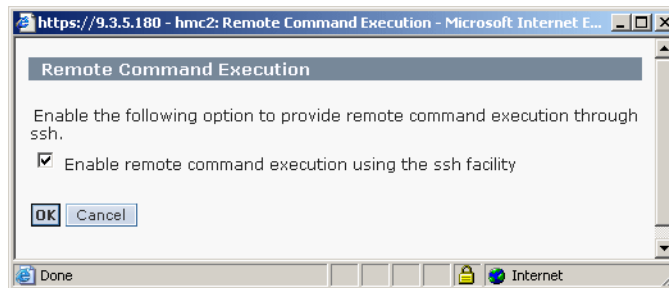


Figure 17-3 Remote command execution window

Use the **mkauthkeys** command in the CLI on either the local or remote HMC to generate SSH authentication key pairs on the local and remote HMC. You must be logged in as a user with **hmcsuperadmin** privileges, such as the **hscroot** user, and authenticate to the remote HMC by using a remote user ID with **hmcsuperadmin** privileges. Authentication to a remote system (in our case, 9.3.5.180) using RSA authentication is displayed in Example 17-1.

Example 17-1 mkauthkeys command for SSH key generation

```
hscroot@hmc1:~> mkauthkeys --ip 9.3.5.180 -u hscroot -t rsa
Enter the password for user hscroot on the remote host 9.3.5.180:

hscroot@hmc1:~> mkauthkeys --test --ip 9.3.5.180 -u hscroot
```

Logical memory block size

Ensure that the logical memory block (LMB) size is the same on the source and destination systems. The default LMB size depends on the amount of memory installed in the CEC. It varies between 16 MB and 256 MB. A change to the LMB size can only be done by a user with the administrator authority, and you must shut down and restart the managed system for the change to take effect.

Figure 17-4 shows how the size of the logical memory block can be modified in the **Performance Setup** menu of the Advanced System Management Interface (ASMI). The ASMI can be launched through the **Operations** section in the task list on the HMC.

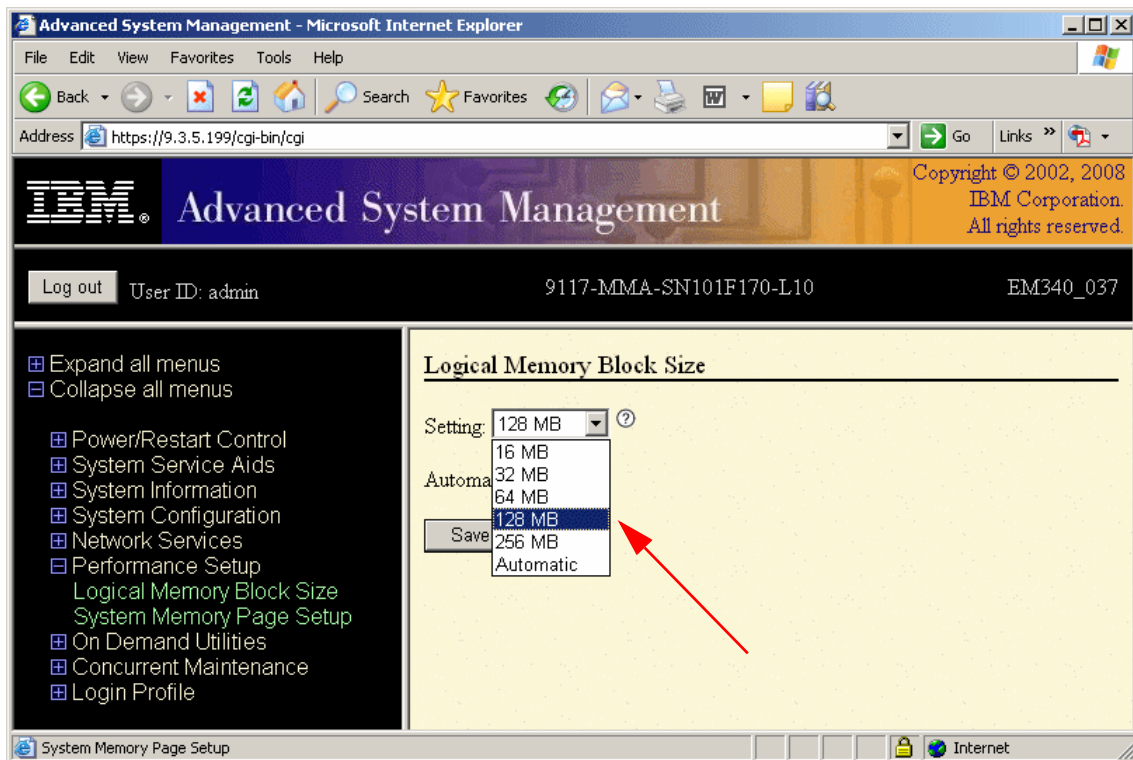


Figure 17-4 Checking and changing LMB size with ASMI

Battery power

Ensure that the destination system is not running on battery power. If the destination system is running on battery power, then you need to return the system to its regular power source before moving a logical partition to it. However, the source system can be running on battery power.

Available memory

Ensure that the destination system has enough available memory to support the mobile partition. To determine the available memory on the destination system, and allocate more memory if necessary, you must have super administrator authority (a user with the HMC hmcsuperadmin role, such as hscroot). The following steps have to be completed on the HMC:

1. Determine the amount of memory of the mobile partition on the source system:
 - a. In the navigation area, open **Systems Management**.
 - b. Select the source system in the navigation area.
 - c. In the contents area, select the mobile partition and select **Properties** in the task list. The Properties window opens.
 - d. Select the **Hardware** tab and then the **Memory** tab.
 - e. View the **Memory** section and record the assigned memory settings.
 - f. Click **OK**.

Figure 17-5 shows the result of the actions.

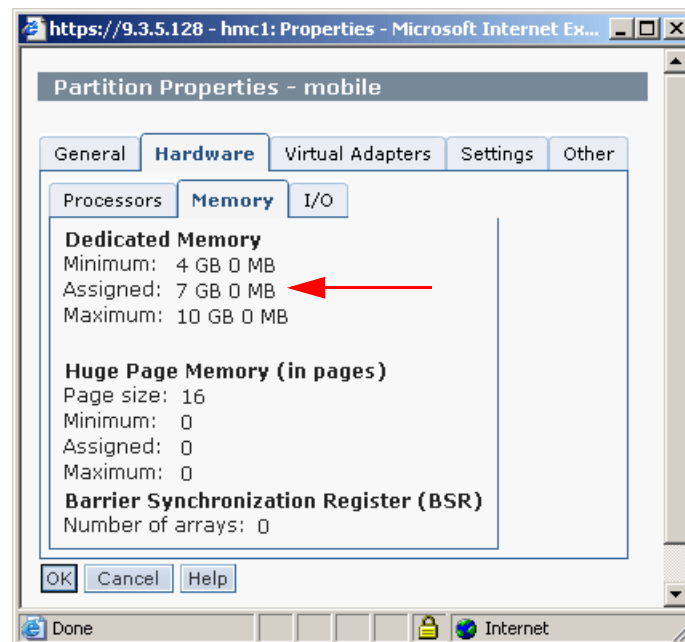


Figure 17-5 Checking the amount of memory of the mobile partition

2. Determine the memory available on the destination system:
 - a. In the contents area, select the destination system and select **Properties** in the task list.
 - b. Select the **Memory** tab.
 - c. Record the Available memory and Current memory available for partition usage.
 - d. Click **OK**.

Figure 17-6 shows the result of the actions.

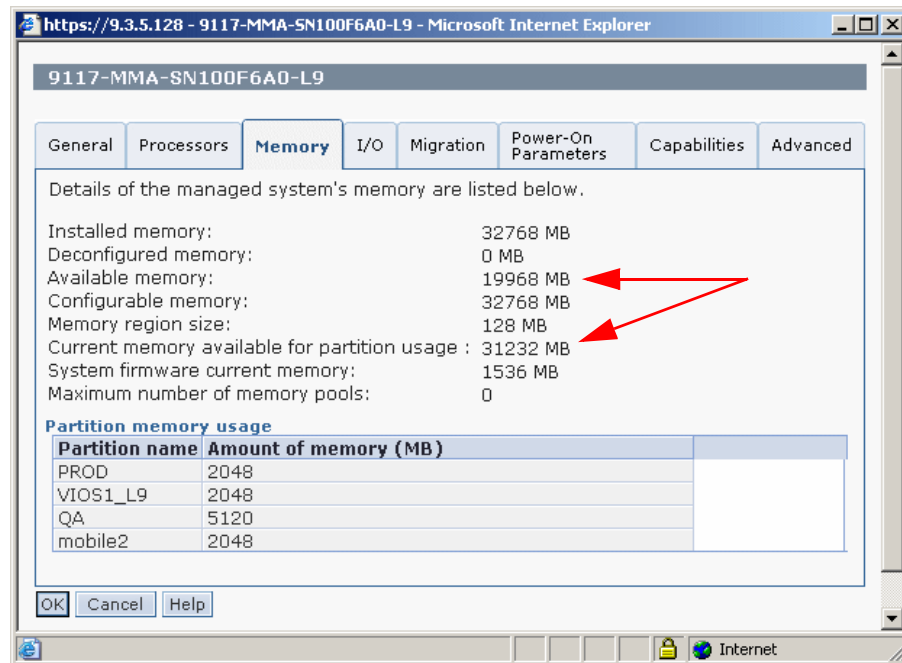


Figure 17-6 Available memory on destination system

3. Compare the values from the previous steps:
 - If the destination system has enough available memory to support the mobile partition, skip the rest of this procedure and continue with other preparation tasks. Note that each partition does require some amount of System firmware memory so you will need to take that into account as well when determining if enough available memory is on the destination side. Note that each partition does require some amount of System firmware memory so you will need to take that into account as well when determining if enough available memory is on the destination side.

- If the destination system does not have enough available memory to support the mobile partition, you must dynamically free up some memory (or use the Capacity on Demand (CoD) feature to activate additional memory, where available) on the destination system before the actual migration can take place. Note, during the migration the HMC will try and steal memory from powered off partitions.

Available processors to support Live Partition Mobility

Ensure that the destination system has enough available processors (or processing units) to support the mobile partition. The profile created on the destination server matches the source server's, therefore dedicated processors must be available on the target if that is what you are using, or enough processing units in a shared processor pool.

To determine the available processors on the destination system and allocate more processors if necessary, you must have super administrator authority (a user with the HMC hmcsuperadmin role, such as hscroot).

Complete the following steps in the HMC:

1. Determine how many processors the mobile partition requires:
 - a. In the navigation area, expand **Systems Management**.
 - b. Select the source system in the navigation area.
 - c. In the contents area, select the mobile partition and select **Properties** in the task list. A new pop-up window called Properties appears.
 - d. Select the **Hardware** tab and then the **Processors** tab.
 - e. View the **Processor** section and record the processing units settings.
 - f. Click **OK**.

Figure 17-7 shows the result of the actions.

Note: In recent HMC levels, p6 appears as POWER6. See Figure 17-7.

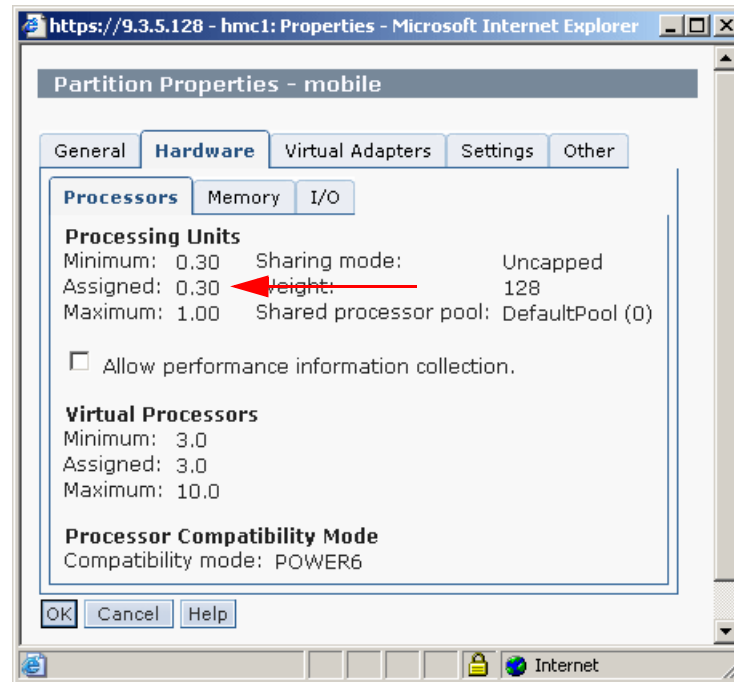


Figure 17-7 Checking the number of processing units of the mobile partition

2. Determine the processors available on the destination system:
 - a. In the contents area, select the destination system and select **Properties** in the task list.
 - b. Select the **Processors** tab.
 - c. Record the Available processors available for partition usage.
 - d. Click **OK**.

Figure 17-8 shows the result of the actions.

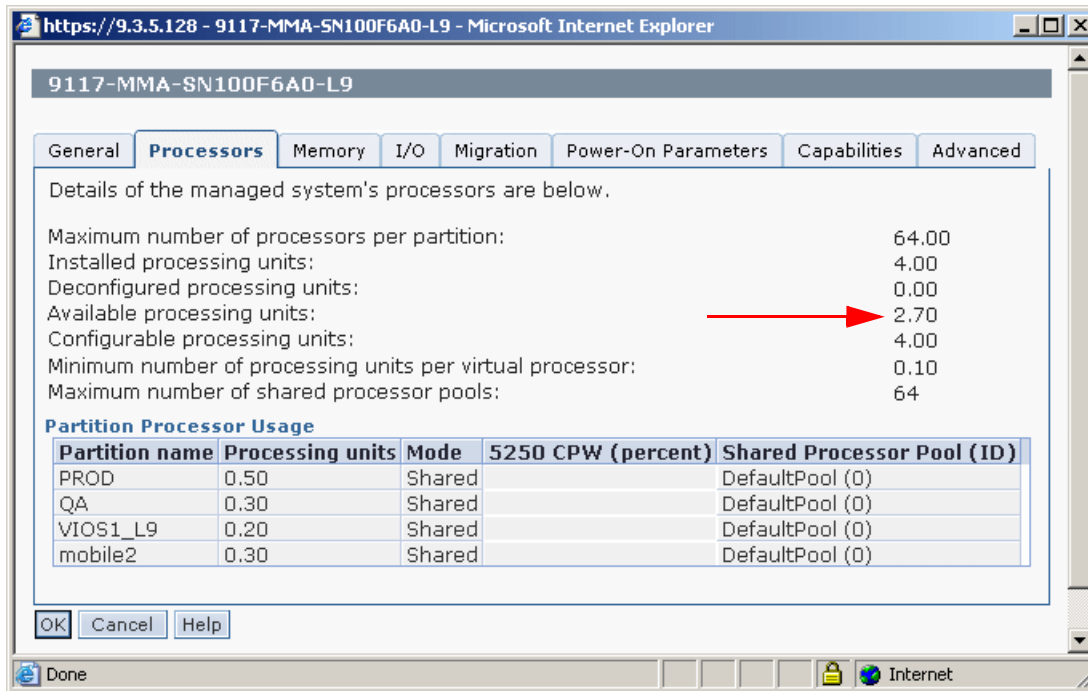


Figure 17-8 Available processing units on destination system

3. Compare the values from the previous steps.
 - If the destination system has enough available processors to support the mobile partition, then skip the rest of this procedure and continue with the remaining preparation tasks for Live Partition Mobility.

If the destination system does not have enough available processors to support the mobile partition, you must dynamically free up processors (or use the CoD feature, when available) on the destination system before the actual migration can take place. Note, during the migration the HMC will try and steal processors from powered off partitions.

Synchronize time-of-day clocks

Another recommended, although optional, task for active partition migration is the synchronization of the time-of-day clocks for the source and destination Virtual I/O Server partitions.

If you choose not to complete this step, the source and destination Virtual I/O Servers synchronize the clocks while the mobile partition is moving from the source system to the destination system. Completing this step before the mobile partition is moved can prevent possible errors.

To synchronize the time-of-day clocks on the source and destination Virtual I/O Servers using the HMC, you must be a super administrator (such as hscroot) to complete the following steps:

1. In the navigation area, open **Systems Management**.
2. Select **Servers** and select the source system.
3. In the contents area, select the source Virtual I/O Server logical partition.
4. Click on **Properties**.
5. Click the **Settings** tab.
6. For Time reference, select **Enabled** and click **OK**.
7. Repeat the previous steps on the destination system for the destination Virtual I/O Server.

Figure 17-9 shows the time-of-day synchronization.

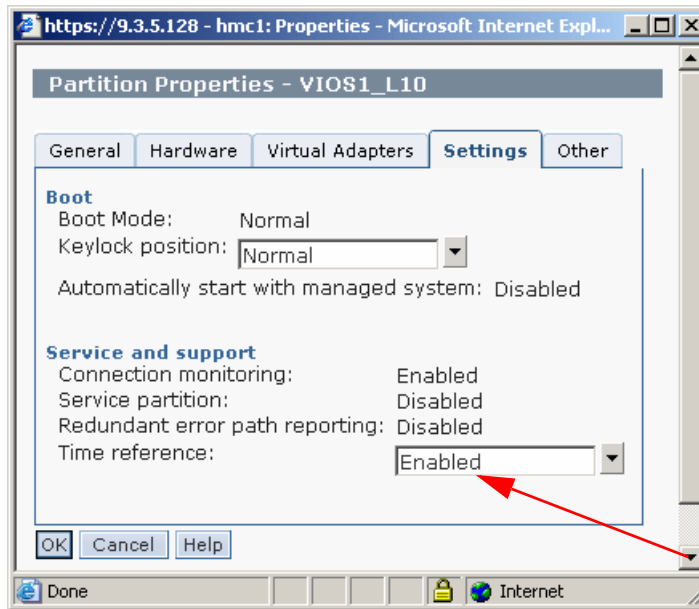


Figure 17-9 Synchronizing the time-of-day clocks

Note: After the Virtual I/O Server infrastructure is configured, a backup of the Virtual I/O Servers is recommended; this approach produces an established checkpoint prior to migration.

RMC connections

Ensure that Resource Monitoring and Control (RMC) connections are established for the Virtual I/O Server mover service partitions on the source and destination system. For *active* partition migration a working RMC connection is also required for the mobile partition except for IBM i which doesn't use RMC and can talk to the POWER Hypervisor directly.

RMC can be configured to monitor resources and perform an action in response to a defined condition. The flexibility of RMC enables you to configure response actions or scripts that manage general system conditions with little or no involvement from the system administrator.

To check the RMC connection status for a Virtual I/O Server, an AIX, or Linux mobile partition run the `lssyscfg` command for the managed system as shown in Example 17-2 and make sure the `rmc_state` is active.

Example 17-2 Checking the RMC connection status

```
hscroot@hmc8:~> lssyscfg -r lpar -F lpar_id,name,state,rmc_state,rmc_ipaddr -m p750
6,IBM i,Running,none,
5,p750_lpar03,Running,active,172.16.21.127
4,p750_lpar02,Running,none,
3,p750_lpar01,Running,active,172.16.21.125
2,p750_vios02,Running,active,172.16.21.112
1,p750_vios01,Running,active,172.16.21.111
```

If the RMC connection is not active, verify and establish the RMC connection specifically for your operating system:

- For AIX or the Virtual I/O Server refer to the IBM Technote *Debug and Fix RMC Connection Errors* at:

<http://www-01.ibm.com/support/docview.wss?uid=isg3T1012915>

- For Linux, install the RSCT utilities. Download these tools from the Service and productivity tools website (and select the appropriate **HMC- or IVM-managed servers** link):

<http://www14.software.ibm.com/webapp/set2/sas/f/1opdiags/home.html>

- Red Hat Enterprise Linux: Install additional software (RSCT Utilities) for Red Hat Enterprise Linux on HMC managed servers.
- SUSE Linux Enterprise Server: Install additional software (RSCT Utilities) for SUSE Linux Enterprise Server on HMC managed servers.

Mover service partition

Ensure that for *active* and *suspended* partition migration at least one of the mover service partitions (MSP) is enabled on a source and destination Virtual I/O Server partition. The mover service partition is a Virtual I/O Server logical partition that is allowed to use its VASI adapter for communicating with the POWER Hypervisor. The mover service partition is not required for *inactive* partition migration.

To enable a Virtual I/O Server as a mover service partition change its partition properties from the HMC to dynamically enable the setting Mover service partition as shown in Figure 17-10.



Figure 17-10 Virtual I/O Server Mover service partition property

Disable redundant error path reporting

Ensure that the mobile partition is not enabled for redundant error path reporting.

Redundant error path reporting allows a logical partition to report server common hardware errors and partition hardware errors to the HMC. Redundant error path reporting must be disabled if you want to migrate a logical partition.

To disable redundant error path reporting for the mobile partition, you must be a super administrator and complete the following steps:

1. In the navigation area, open **Systems Management**.
2. Select **Servers** and select the source system.
3. In the contents area, select the logical partition you wish to migrate and select **Configuration** → **Manage Profiles**.
4. Select the active logical partition profile and select **Edit** from the **Actions** menu.
5. Click the **Settings** tab.

6. Deselect **Enable redundant error path reporting**, and click **OK**.
7. Because disabling redundant error path reporting cannot be done dynamically, you have to shut down the mobile partition, then power it on using the profile with the modifications.

Figure 17-11 shows the disabled redundant error path handling:

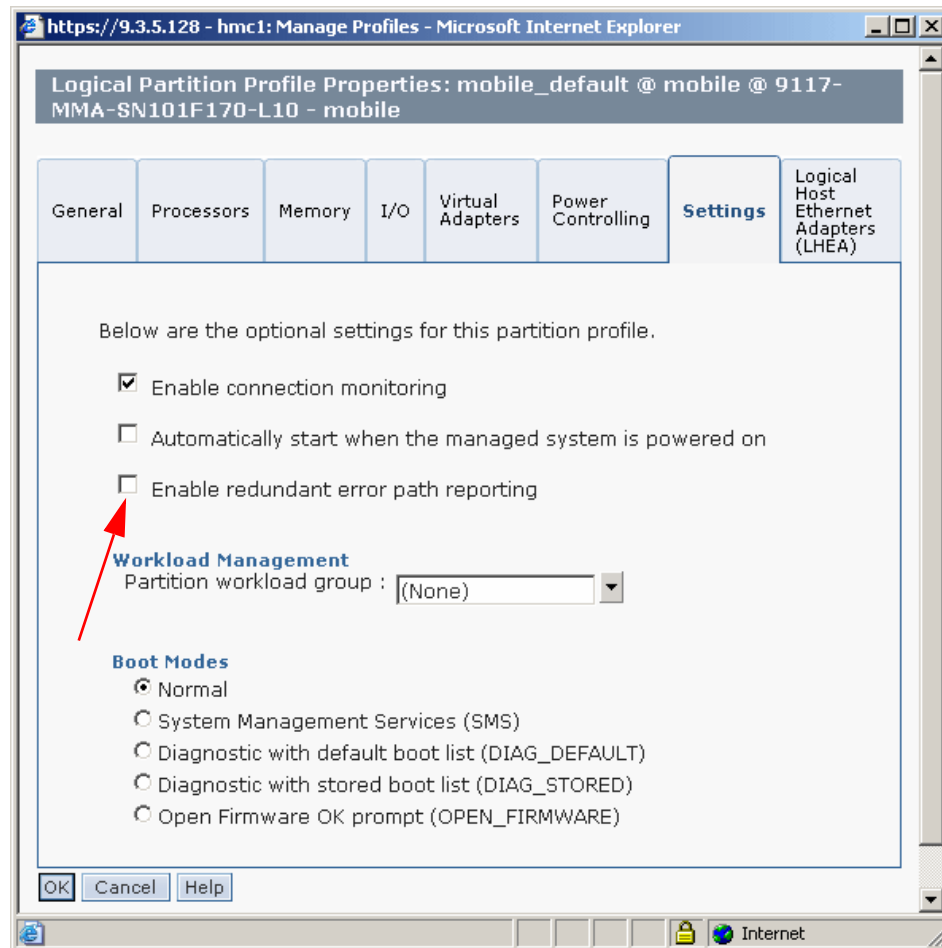


Figure 17-11 Disable redundant error path handling

Virtual serial adapters

Ensure that the mobile partition is not using a virtual serial adapter in slots higher than slot 1.

Virtual serial adapters are often used for virtual terminal connections to the operating system. The first two virtual serial adapters (slots 0 and 1) are reserved for the HMC. For a logical partition to participate in a partition migration, it cannot have any required virtual serial adapters, except for the two reserved for the HMC.

Note: Any open virtual TTY terminal or console sessions are disconnected during the migration and can be newly opened on the destination system by the user if desired.

To dynamically disable unreserved virtual serial adapters using the HMC, you must be a super administrator and complete the following steps:

1. In the navigation area, expand **Systems Management**.
2. Select **Servers** and select the source system.
3. In the contents area, select the logical partition to migrate and select **Configuration** → **Manage Profiles**.
4. Select the active logical partition profile and select **Edit** from the **Actions** menu.
5. Select the **Virtual Adapter** tab.
6. If there are more than two virtual serial adapters listed, then ensure that the adapters in slots 2 and higher are not selected as Required.
7. Click **OK**.

Figure 17-12 shows the result of the steps.

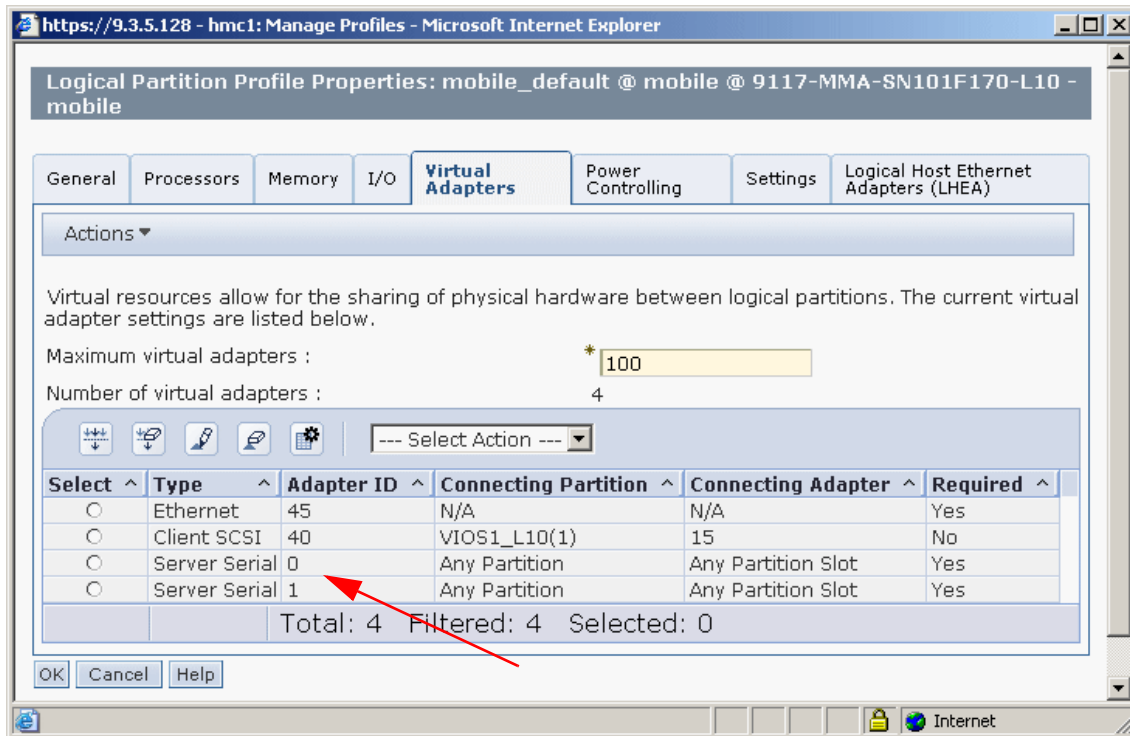


Figure 17-12 Verifying the number of serial adapters on the mobile partition

Partition workload groups

Ensure that the mobile partition is not part of a logical partition group.

A partition workload group identifies a set of partitions that reside on the same system. The partition profile specifies the name of the partition workload group to which it belongs, if applicable. For a logical partition to participate in a partition migration, it cannot be assigned to a partition workload group.

To dynamically remove the mobile partition from a partition workload group, you must be a super administrator on the HMC and complete the following steps:

1. In the navigation area, expand **Systems Management** → **Servers**.
2. In the contents area, open the source system.
3. Select the mobile partition and select **Properties**.
4. Click the **Other** tab.
5. In the **Workload group** field, select **(None)**.
6. In the contents area, open the mobile partition and select **Configuration** → **Manage Profiles**.
7. Select the active logical partition profile and select **Edit** from the **Actions** menu.
8. Click the **Settings** tab.
9. In the **Workload Management** area, select **(None)** and click **OK**.
10. Repeat the last three steps for all partition profiles associated with the mobile partition.

Figure 17-13 and Figure 17-14 show the tabs for the disablement of the partition workload group (both in the partition and in the partition profiles).

Figure 17-13 shows the Other tab.

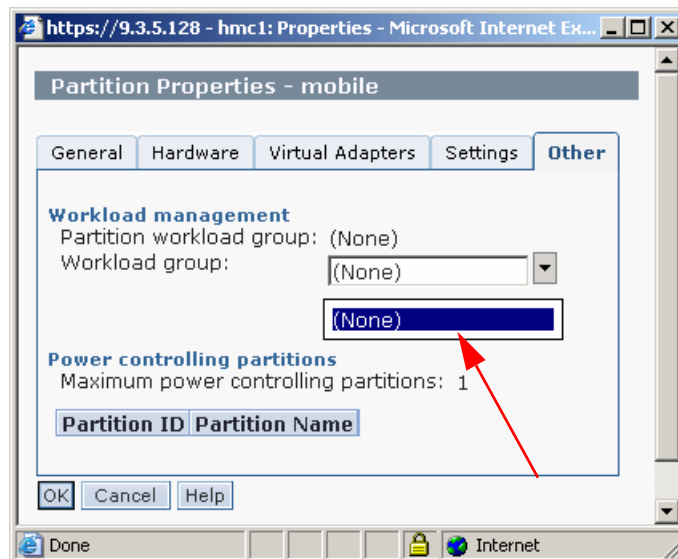


Figure 17-13 Disabling partition workload group - Other tab

Figure 17-14 shows the Settings tab.

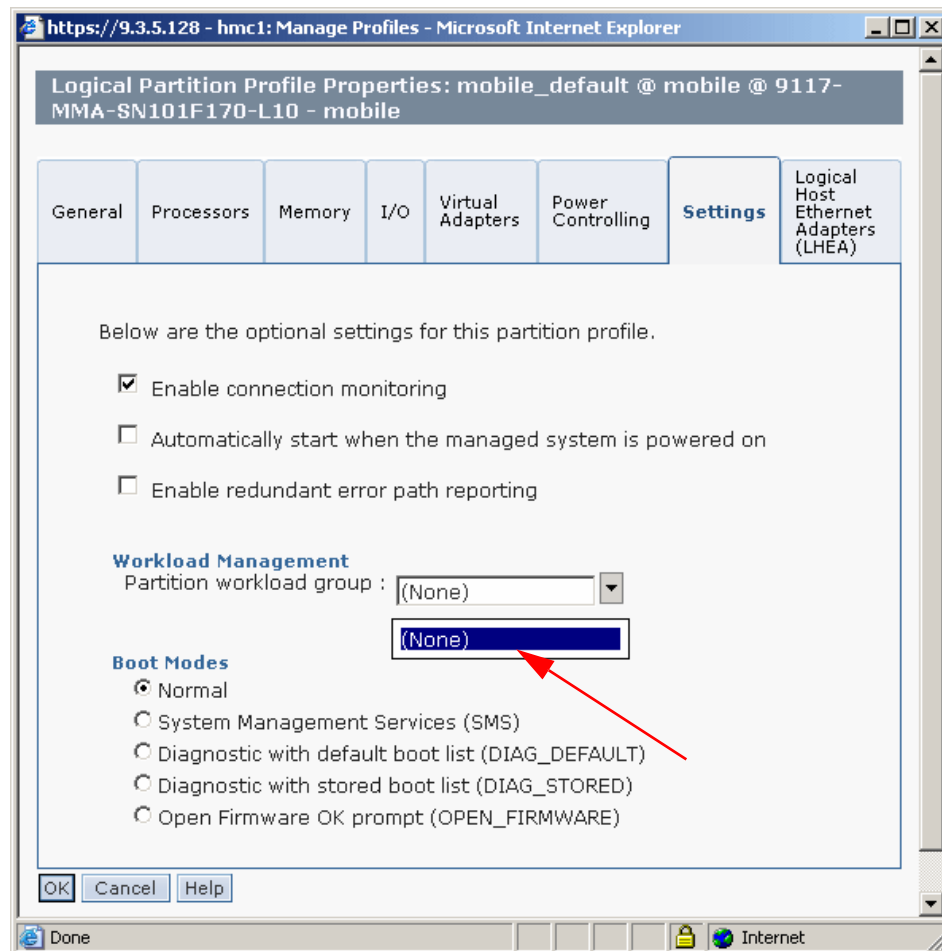


Figure 17-14 Disabling partition workload group - Settings tab

Barrier-synchronization register

Ensure that the mobile partition is not using barrier-synchronization register (BSR) arrays.

BSR is a memory register that is located on certain POWER technology-based processors. A parallel-processing application running on AIX can use a BSR to perform barrier synchronization, which is a method for synchronizing the threads in the parallel-processing application. For a logical partition to participate in active partition migration, it cannot use BSR arrays. However, it can still participate in inactive partition migration if it uses BSR.

To disable BSR for the mobile partition using the HMC, you must be a super administrator and complete the following steps:

1. In the navigation area, expand **Systems Management**.
 2. Select **Servers**.
 3. In the contents area, open the source system.
 4. Select the mobile partition and select **Properties**.
 5. Click the **Hardware** tab.
 6. Click the **Memory** tab.
- If the number of BSR arrays equals zero, the mobile partition can participate in inactive or active migration, as shown in Figure 17-15. You can now continue with additional preparatory tasks for the mobile partition.

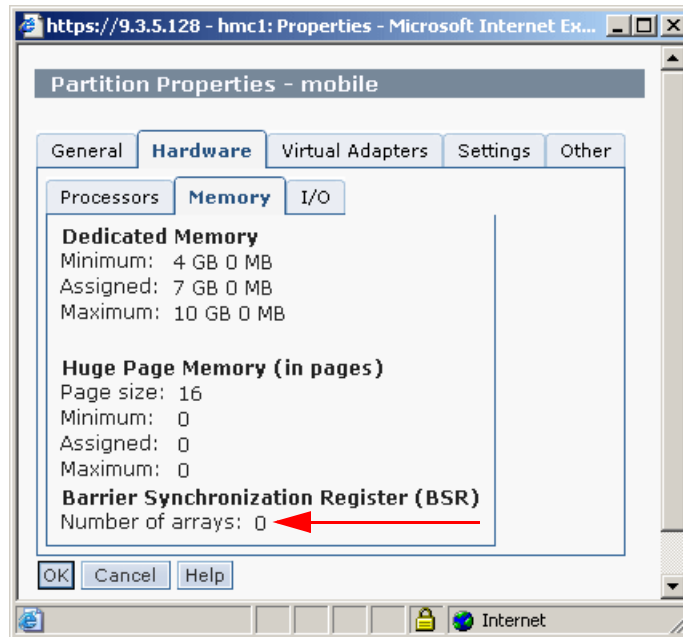


Figure 17-15 Checking the number of BSR arrays on the mobile partition

- If the number of BSR arrays is not equal to zero, take one of the following actions:
 - Perform an inactive migration instead of an active migration. Skip the remaining steps and see 11.2.4, “Migratability” on page 263.
 - Click **OK** and continue to the next step to prepare the mobile partition for an active migration.

7. In the contents area, open the mobile partition and select **Configuration** → **Manage Profiles**.
8. Select the active logical partition profile and select **Edit** from the **Actions** menu.
9. Click the **Memory** tab.
10. Enter 0 in the BSR arrays for this profile field and click **OK** (see Figure 17-16).
11. Because modifying BSR cannot be done dynamically, you have to shut down the mobile partition, then power it on by using the profile with the BSR modifications.

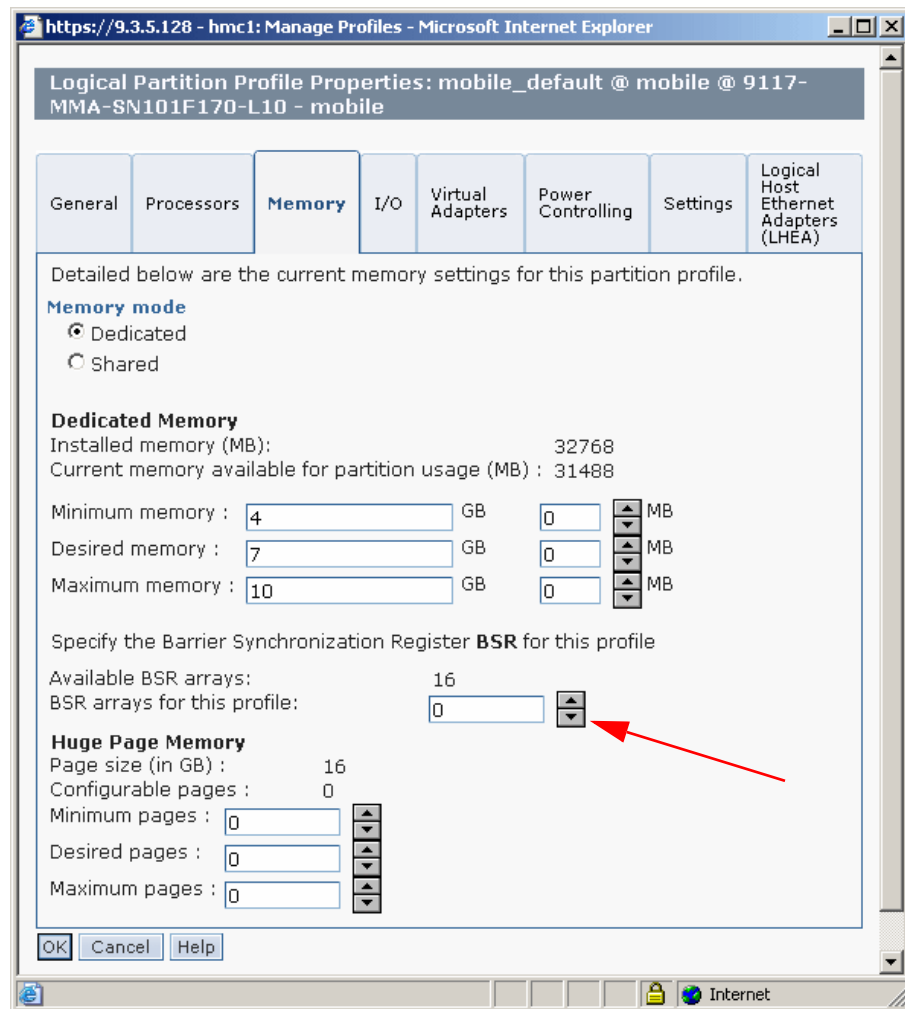


Figure 17-16 Setting number of BSR arrays to zero

Huge pages

Ensure that the mobile partition is not using huge pages.

Huge pages can improve performance in specific environments that require a high degree of parallelism, such as in DB2 partitioned database environments. You can specify the minimum, desired, and maximum number of huge pages to assign to a partition when you create a partition profile. For a logical partition to participate in active partition migration, it cannot use huge pages. However, if the mobile partition does use huge pages, it can still participate in inactive partition migration.

To configure huge pages for the mobile partition using the HMC, you must be a super administrator and complete the following steps:

1. Open the source system and select **Properties**.
2. Click the **Advanced** tab.
 - If the current huge page memory equals zero (0), shown in Figure 17-17, skip the remaining steps of this procedure and continue with additional preparatory tasks for the mobile partition.

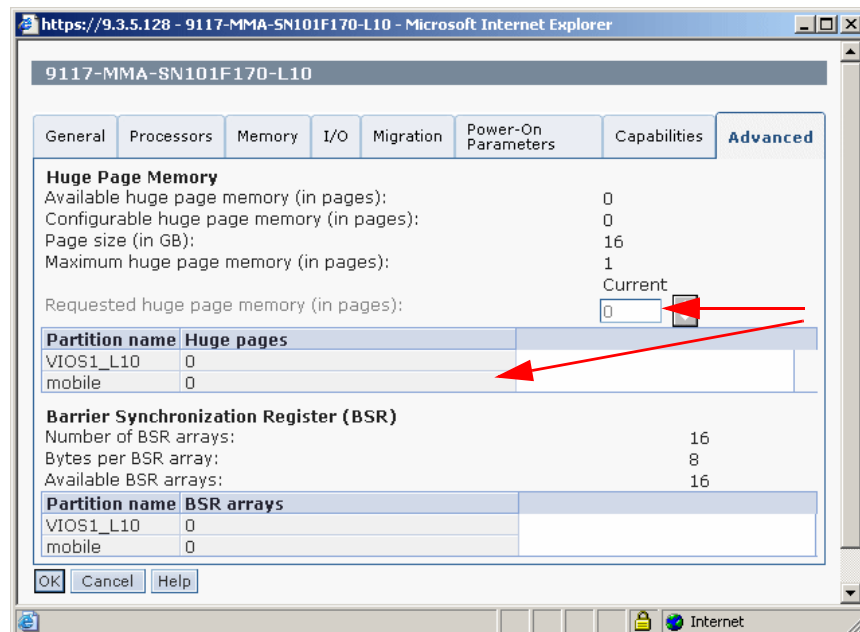


Figure 17-17 Checking if huge page memory equals zero

- If the current huge page memory is not equal to 0, take one of the following actions:
 - Perform an inactive migration instead of an active migration. Skip the remaining steps and see 11.2.4, “Migratability” on page 263.
 - Click **OK** and continue with the next step to prepare the mobile partition for an active migration.
- 3. In the contents area, open the mobile partition and select **Configuration** → **Manage Profiles**.
- 4. Select the active logical partition profile and select **Edit** from the **Actions** menu.
- 5. Click the **Memory** tab.

6. Enter 0 in the field for desired huge page memory; click **OK** (Figure 17-18).
7. Because changing huge pages cannot be done dynamically, shut down the mobile partition, then turn it on by using the profile with the modifications.

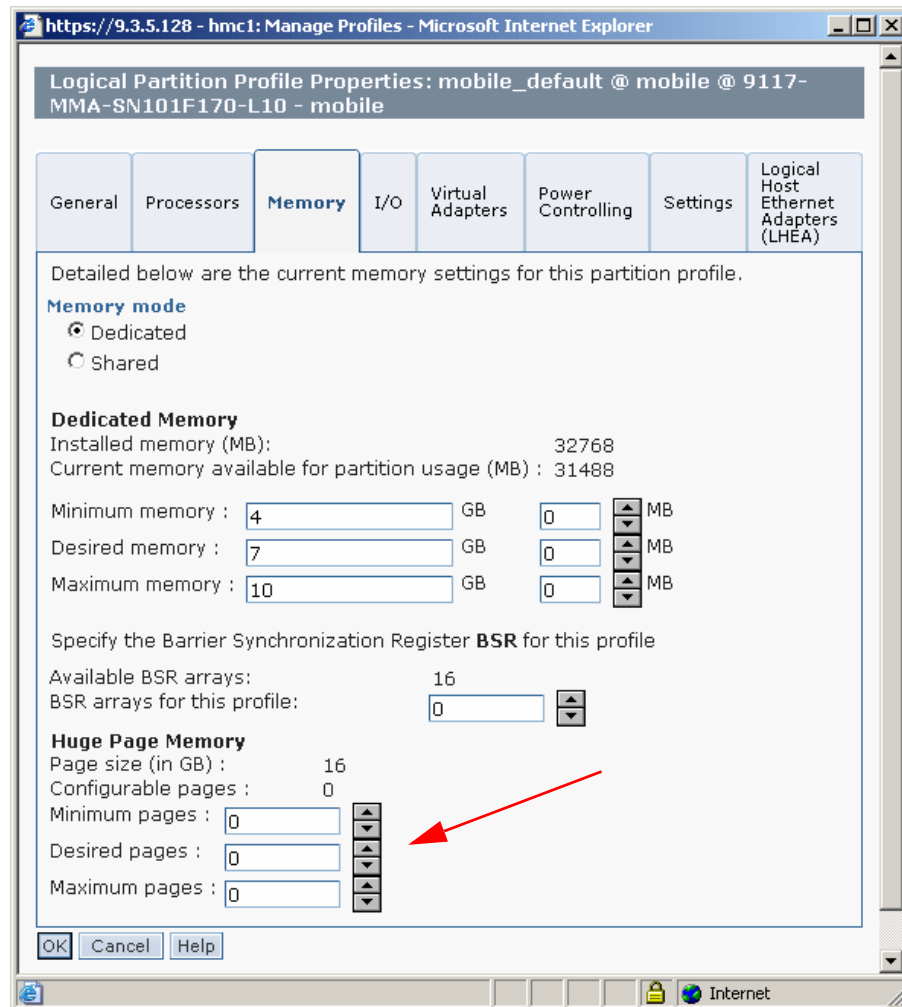


Figure 17-18 Setting Huge Page Memory to zero

Configuring an IBM i mobile partition for restricted I/O

Ensure that the IBM i partition to be migrated is set to “Restricted IO Partition” in its partition properties like shown in Figure 17-19. Note that this setting can only be changed when the IBM i partition is not activated and enabling the restricted I/O setting requires prior removal of any physical I/O adapters from the partition profile.

Figure 17-19 shows the restricted I/O setting selected.

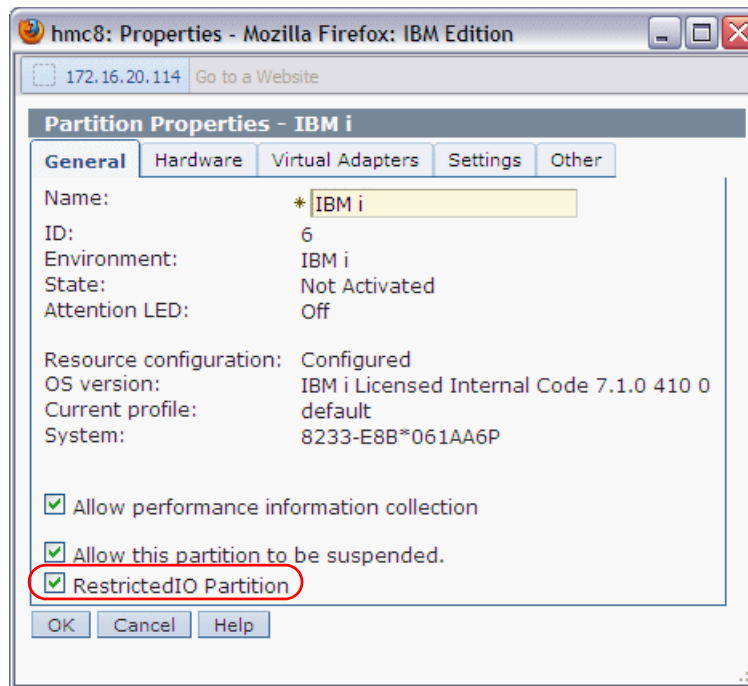


Figure 17-19 IBM i partition property restricted IO setting

17.1.3 Configuring the external storage

This section describes the tasks that you must complete to ensure your storage configuration meets the minimal configuration for Live Partition Mobility before you can actually migrate your logical partition.

To configure external storage:

1. If using NPIV, ensure that:
 - a. Both virtual WWPN addresses of the virtual Fibre Channel client adapters are correctly zoned and configured with regards to LUN mapping in the SAN storage environment
 - b. The backing physical FC adapters of the source and destination Virtual I/O Servers are attached to the corresponding SAN storage subsystems
2. If using virtual SCSI, ensure that:
 - a. The same SAN disks used as virtual SCSI disks by the mobile partition are assigned to source and destination Virtual I/O Server logical partitions.

Note: The `hdisk` resource names for the SAN disks don't necessarily match between different Virtual I/O Servers. Use the *unique device identifier* (UDID) or *IEEE volume identifier* from the `lsdev -dev hdiskX -attr` command output to correlate matching of the same SAN disks between the Virtual I/O Servers on the source and destination system.

- b. the `reserve_policy` attributes on the shared physical volumes is set to `no_reserve` on the source and destination Virtual I/O Servers:
 - To list all the disks, type the following command:
`lsdev -type disk`
 - To list the attributes of `hdiskX`, type the following command:
`lsdev -dev hdiskX -attr`
 - If `reserve_policy` is not set to `no_reserve`, use the following command:
`chdev -dev hdiskX -attr reserve_policy=no_reserve`
- c. the physical volumes on the source and destination Virtual I/O Servers have the same unique identifier, physical identifier, or an IEEE volume attribute. These identifiers are required in order to export a physical volume as a virtual device.

To list disks with a *unique device identifier* (UDID):
 - i. Type the `oem_setup_env` command on the Virtual I/O Server CLI.
 - ii. Type the `odmget -qattribute=unique_id CuAt` command to list the disks that have a UDID. See Example 17-3.

Example 17-3 Output of `odmget` command

CuAt:

```
name = "hdisk6"
attribute = "unique_id"
value = "3E213600A0B8000291B080000520E023C6B8D0F1815"    FASSt03IBMfcp"
type = "R"
generic = "D"
rep = "n1"
nls_index = 79
```

CuAt:

```
name = "hdisk7"
attribute = "unique_id"
value = "3E213600A0B8000114632000073244919ADCA0F1815"    FASSt03IBMfcp"
```

```
type = "R"
generic = "D"
rep = "n1"
nls_index = 79
```

iii. Type **exit** to return to the Virtual I/O Server prompt.

To list disks with a *physical identifier* (PVID):

- i. Type the **lspv** command to list the devices with a PVID. See Example 17-4. If the second column has a value of none, the physical volume does not have a PVID. Unless using the physical devices in a Shared Storage Pool the recommendation is to put a PVID on the physical volume before it is exported as a virtual device.

Example 17-4 Output of lspv command

\$ lspv			
NAME	PVID	VG	STATUS
hdisk0	00c1f170d7a97dec	rootvg	active
hdisk6	00c0f6a0915fc126	None	
hdisk7	00c0f6a08de5008b	None	

- ii. Type the **chdev** command to put a PVID on the physical volume in the following format:

```
chdev - dev physicalvolumename -attr pv=yes -perm
```

To list disks with an *IEEE volume attribute identifier* like for DS4000 storage volumes using the older RDAC multipath driver instead of newer default MPIO, issue the following command (in the shell `oem_setup_env`):

```
odmget -qattribute=ieee_volname CuAt
```

3. Verify that the destination Virtual I/O Server has sufficient free virtual slots to create the virtual SCSI or Fibre Channel adapters required to host the mobile partition. To verify the virtual SCSI configuration using the HMC, you must be a super administrator (such as `hscroot`) to complete the following steps:
 - a. In the navigation area, open **Systems Management**.
 - b. Select **Servers**.
 - c. In the contents area, open the destination system.
 - d. Select the destination Virtual I/O Server logical partition and click **Properties**.

- e. Select the **Virtual Adapters** tab and compare the number of virtual adapters to the maximum virtual adapters. This is shown in Figure 17-20,

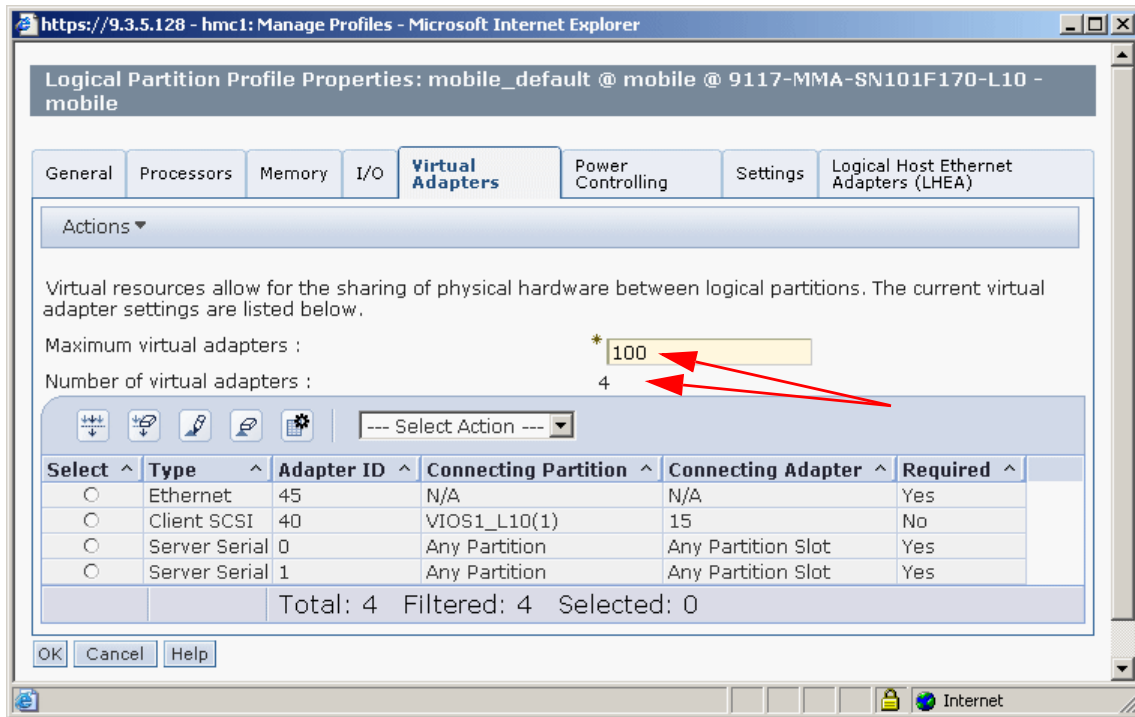


Figure 17-20 Checking free virtual slots

- If, after verification, the number of maximum virtual adapters is higher or equal to the number of virtual adapters plus the number of virtual SCSI or/and virtual Fibre Channel adapters required to host the migrating partition, you can continue with additional preparatory tasks at step 4 on page 648.
- If the maximum virtual adapter value does not allow the addition of required virtual adapters for the mobile partition, then you have to modify its partition profile by completing the following steps:
 - i. In the navigation area, open **Systems Management**.
 - ii. Select **Servers**.
 - iii. In the contents area, open the destination system.
 - iv. Select the destination Virtual I/O Server logical partition.
 - v. Click in the task area on configuration, click **Manage profiles**.

- vi. Select the active logical partition profile and select **Edit** from the **Actions** menu.
 - vii. Click the **Virtual Adapters** tab and modify (increase) the number of maximum virtual adapters. You must shut down and restart the logical partition for the change to take effect.
4. Verify that the AIX or Linux mobile partition does not have physical or required I/O adapters and devices – for an IBM i partition this is prevented anyway by the required Restricted IO Partition property setting. This is only an issue for active partition migration. If you want to perform an active migration, you must move the physical or required I/O from the mobile partition.
 5. All profile changes on the mobile partition's profile must be activated before starting the migration so that the new values can take effect:
 - a. If the partition is not activated, it must be powered on. It is sufficient to activate the partition to the SMS menu.If the partition is active, you can shut it down and power on the partition again by using the changed logical partition profile.

17.2 Suspend and Resume setup

This section will tell you setup details required in a PowerVM to use this capability.

17.2.1 Creating a reserved storage device pool

This section shows how to create the reserved storage device pool using the HMC. The creation of the reserved storage device pool is required in order to use the Partition Suspend and Resume capability on a PowerVM Standard Edition environment, or in a PowerVM Enterprise Edition environment where Active Memory Sharing has not been configured.

Attention: When configuring Active Memory Sharing, the reserved storage device pool gets automatically created when creating a shared memory pool.

Follow these steps:

1. On the HMC, select the managed system on which the reserved storage device pool must be created, then select **Configuration** → **Virtual Resources** → **Reserved Storage Device Pool Manager**, as shown in Figure 17-21.

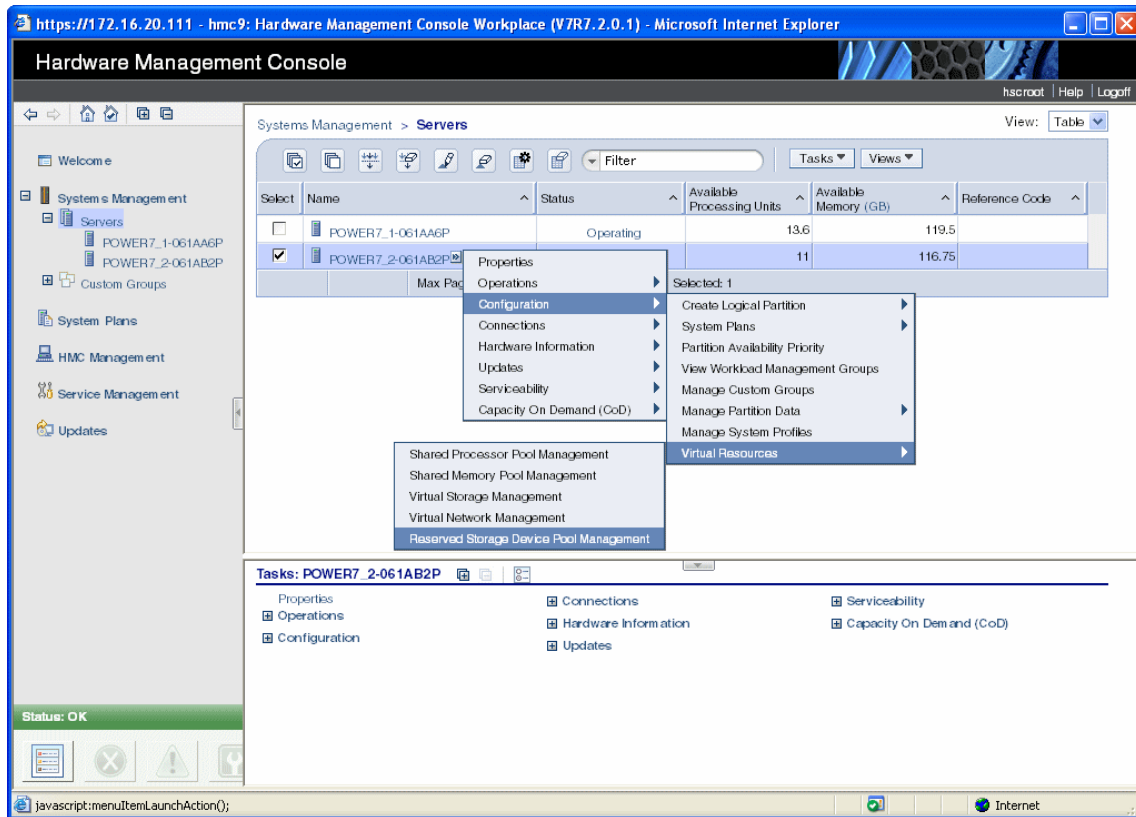


Figure 17-21 Reserved storage device pool management access menu

2. Select the Virtual I/O Server on which the reserved storage device pool must be created, as shown in Figure 17-22, then click **Select Devices**.

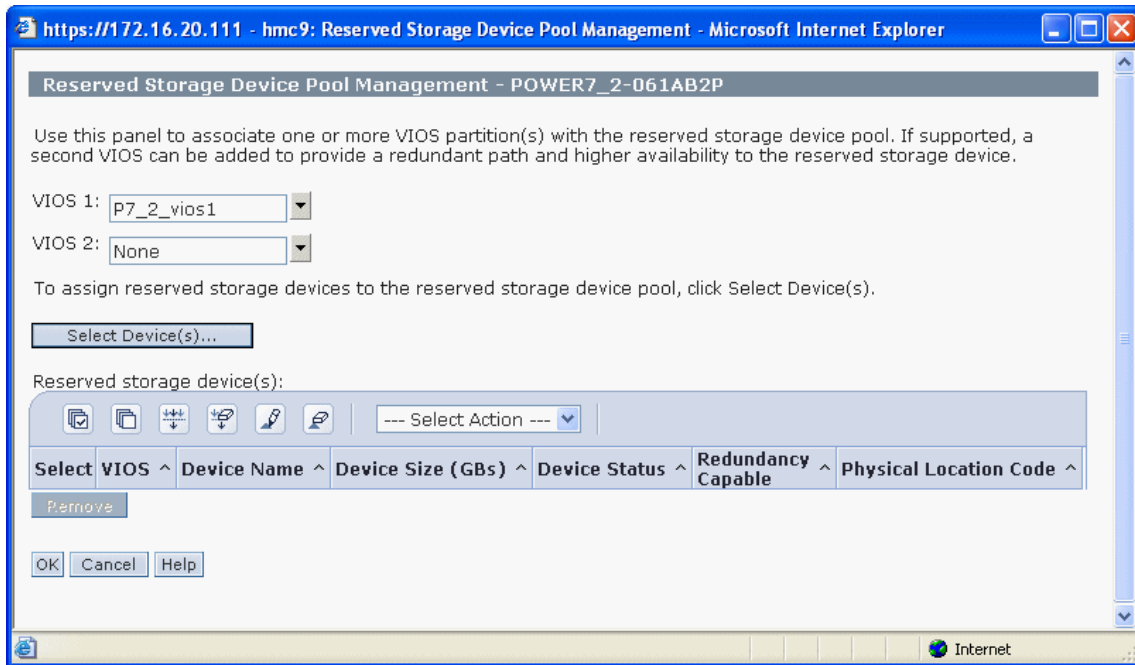


Figure 17-22 Reserved storage device pool management

Tip: A second Virtual I/O Server with access to the reserved storage device can be selected to increase availability.

3. Select the device type, optionally select the minimum and maximum device size, as shown in Figure 17-23, then click **Refresh** to display the list of available devices.

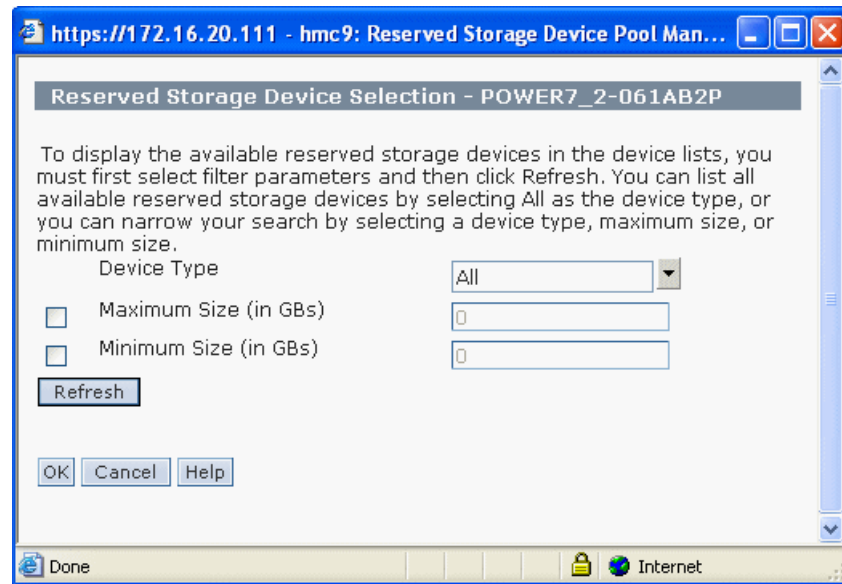


Figure 17-23 Reserved storage device list selection

4. Select the devices to add to the reserved storage device pool, as shown in Figure 17-24, then click **OK**.

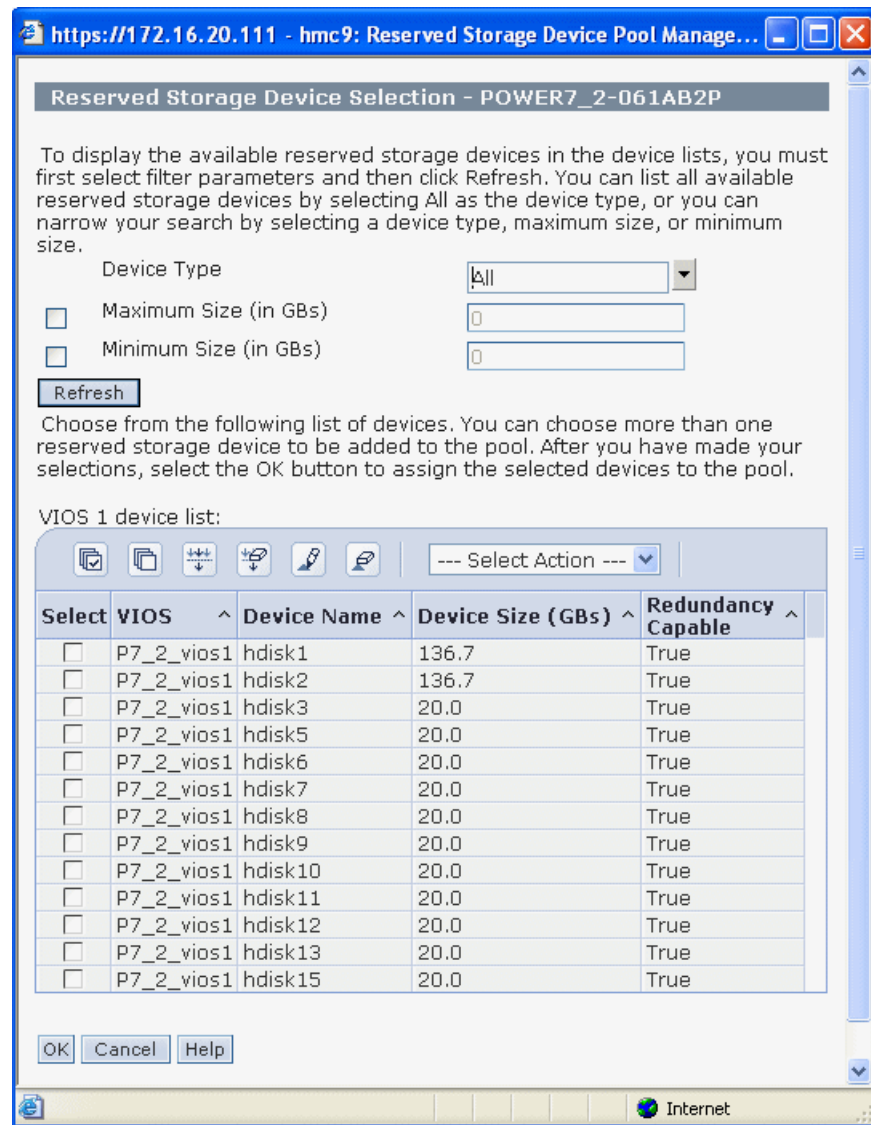


Figure 17-24 Reserved storage device selection

- Review the Virtual I/O Server and devices selected (Figure 17-25), then click **OK** to proceed with the reserved storage pool device creation.

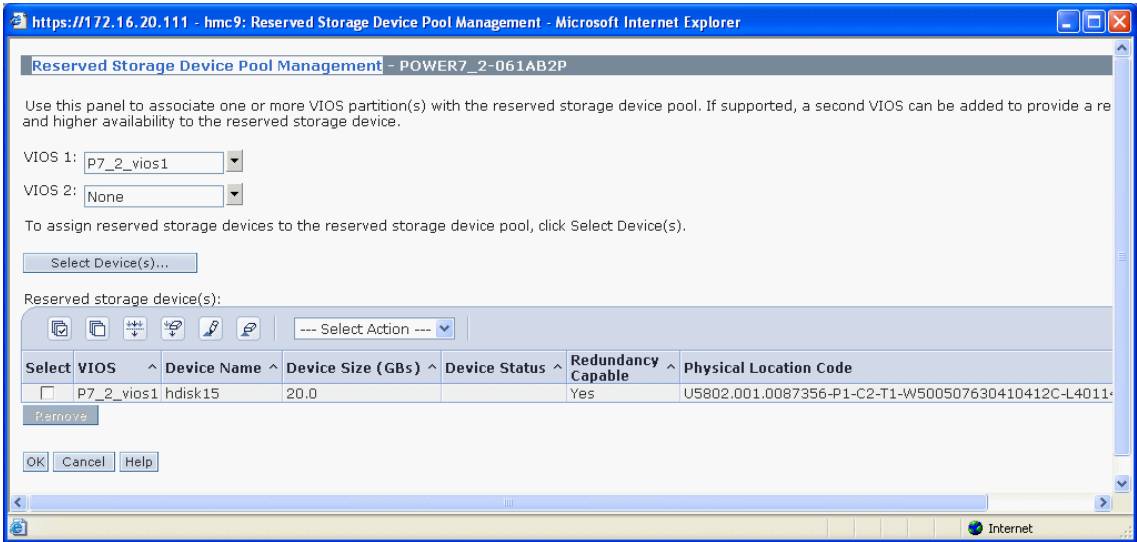


Figure 17-25 Reserved storage device pool creation

After the reserved storage device pool has been created, as for the shared memory pool, four new VASI and VBSD devices, as well as one VRMPAGE device for each device in the reserved storage device pool, will be visible on the Virtual I/O Server, as shown in Example 17-5.

Example 17-5 Listing VASI, VBSD and VRMPAGE devices

```
$ lsdev -dev vasi*
name          status      description
vasi0         Available  Virtual Asynchronous Services Interface (VASI)
vasi1         Available  Virtual Asynchronous Services Interface (VASI)
vasi2         Available  Virtual Asynchronous Services Interface (VASI)
vasi3         Available  Virtual Asynchronous Services Interface (VASI)
vasi4         Available  Virtual Asynchronous Services Interface (VASI)
$ lsdev -dev vbsd*
name          status      description
vbsd0         Available  Virtual Block Storage Device (VBSD)
vbsd1         Available  Virtual Block Storage Device (VBSD)
vbsd2         Available  Virtual Block Storage Device (VBSD)
vbsd3         Available  Virtual Block Storage Device (VBSD)
vbsd4         Available  Virtual Block Storage Device (VBSD)
$ lsdev -dev vrmpage*
name          status      description
vrmpage0      Defined    Paging Device - Disk
```

From the HMC command line interface, the command **lshwres** can be used to display the reserved storage device pool configuration, as shown in Example 17-6.

Example 17-6 Displaying reserved storage device pool using lshwres

```
hscroot@hmc9:~> lshwres -r rspool -m POWER7_2-061AB2P  
vios_names=P7_2_vios1,vios_ids=1
```

When using the **lshwres** command with the **--rsubtype rsdev** flag, as shown in Example 17-7, the attributes and status of each device in the reserved storage device pool are displayed.

Example 17-7 Displaying devices in the reserved storage device pool using lshwres

```
hscroot@hmc9:~> lshwres -r rspool -m POWER7_2-061AB2P --rsubtype rsdev  
device_name=hdisk15,vios_name=P7_2_vios1,vios_id=1,size=20480,type=phys  
,state=Inactive,phys_loc=U5802.001.0087356-P1-C2-T1-W500507630410412C-L  
4011401200000000,is_redundant=0,lpar
```

Devices: In an Active Memory Sharing configuration, devices used as paging devices will be listed when displaying the devices in the reserved storage device pool, and equally, devices in the reserved storage device pool will be listed when displaying paging devices.

17.2.2 Creating a suspend and resume capable partition

The process to create a suspend and resume capable partition is almost identical as the process to create any other partition. The only difference resides in the **Create Partition** section of the **Create Lpar Wizard**, where the check box **Allow this partition to be suspended**, as shown in Figure 17-26, must be selected.

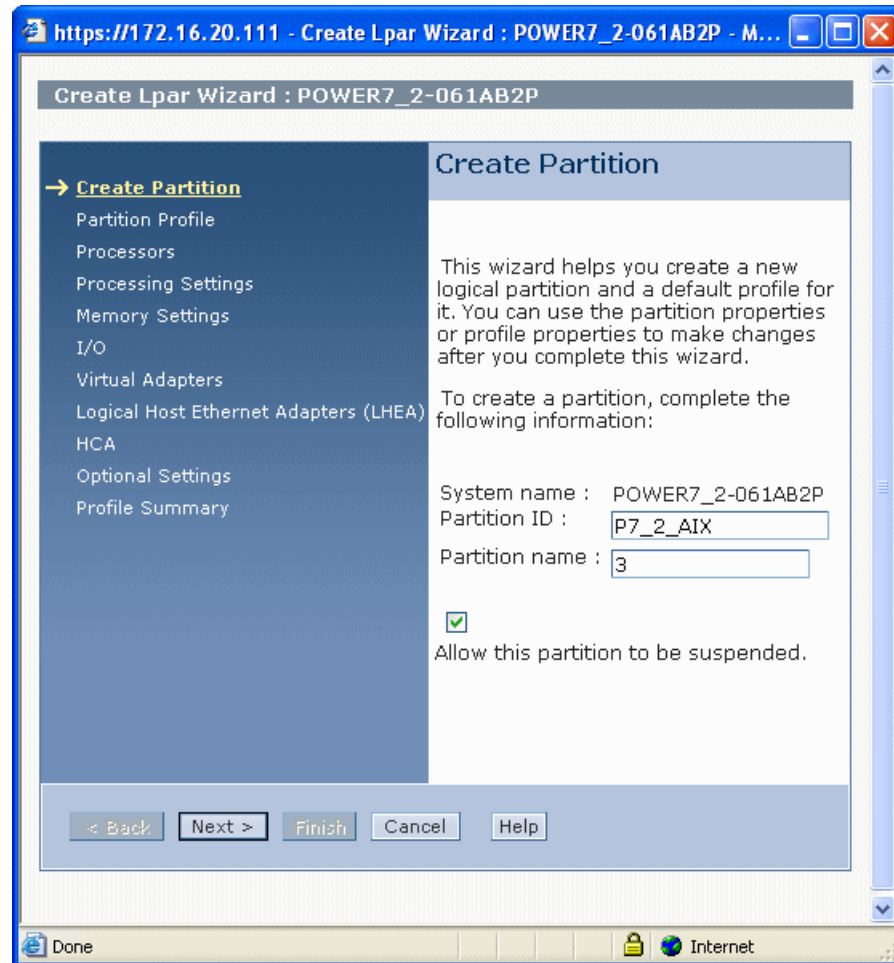


Figure 17-26 Creating a suspend and resume capable partition

Considerations:

- ▶ The **Allow this partition to be suspended** option is not enabled by default when using the HMC GUI **Create Lpar Wizard**.
- ▶ The parameter **suspend_capable** is optional when using the **mksyscfg** command from the HMC command line interface to define a new partition.

For active partitions that were created without enabling **Allow this partition to be suspended**, or for active partitions that were created with a Hardware Management Console version lower than 7.7.2.0, you can enable this feature while the partition is running.

From the Hardware management Console, perform the following steps:

1. Select the partition.
2. Select **Properties**.
3. In the **General** panel, mark the check box **Allow this partition to be suspended**.

From the Hardware Management Console command line interface, you can use the **chsyscfg** command, as shown in Example 17-8, to enable the partition suspend capability.

Example 17-8 Enabling the resume and suspend capability with the chsyscfg command

```
hscroot@hmc9:~> chsyscfg -r lpar -m POWER7_2-061AB2P -i  
"name=P7_2_AIX,suspend_capable=1"  
hscroot@hmc9:~>
```

Validation: When modifying the `suspend_capable` attribute of a running partition, the Hardware Management Console validates resources and settings restrictions before applying the change.

If you want to verify that a partition is suspend and resume capable enabled, from the Hardware Management Console command line interface, you can use the `lssyscfg` command, as shown in Example 17-9.

Example 17-9 Display the partition attribute with the `lssyscfg` command

```
hscroot@hmc9:~> lssyscfg -r lpar -m POWER7_2-061AB2P --filter
"lpar_names=P7_2_AIX"
name=P7_2_AIX,lpar_id=3,lpar_env=aixlinux,state=Running,resource_config
=1,os_version=AIX 7.1
7100-00-00-0000,logical_serial_num=061AB2P3,default_profile=default,cur
r_profile=default,work_group_id=none,shared_proc_pool_util_auth=0,allow
_perf_collection=0,power_ctrl_lpar_ids=none,boot_mode=norm,lpar_keylock
=norm,auto_start=0,redundant_err_path_reporting=0,rmc_state=active,rmc_
ipaddr=172.16.20.172,time_ref=0,lpar_avail_priority=127,desired_lpar_pr
oc_compat_mode=default,curr_lpar_proc_compat_mode=POWER7,suspend_capabl
e=1
```

17.2.3 Validating that a partition is suspend capable

The suspend validation process verifies that a given partition meets all the requirements to be suspended and that no major configuration issue can prevent this operation from being performed successfully.

The validation process is included in the suspend and resume operation wizard, therefore, you have the possibility to perform a validation prior to executing the actual suspend operation.

This section shows how to perform the suspend validation operation using the HMC:

1. On the HMC, select the partition you want the suspend validation to be performed on, then select **Operations** → **Suspend Operations** → **Suspend** as shown in Figure 17-27.

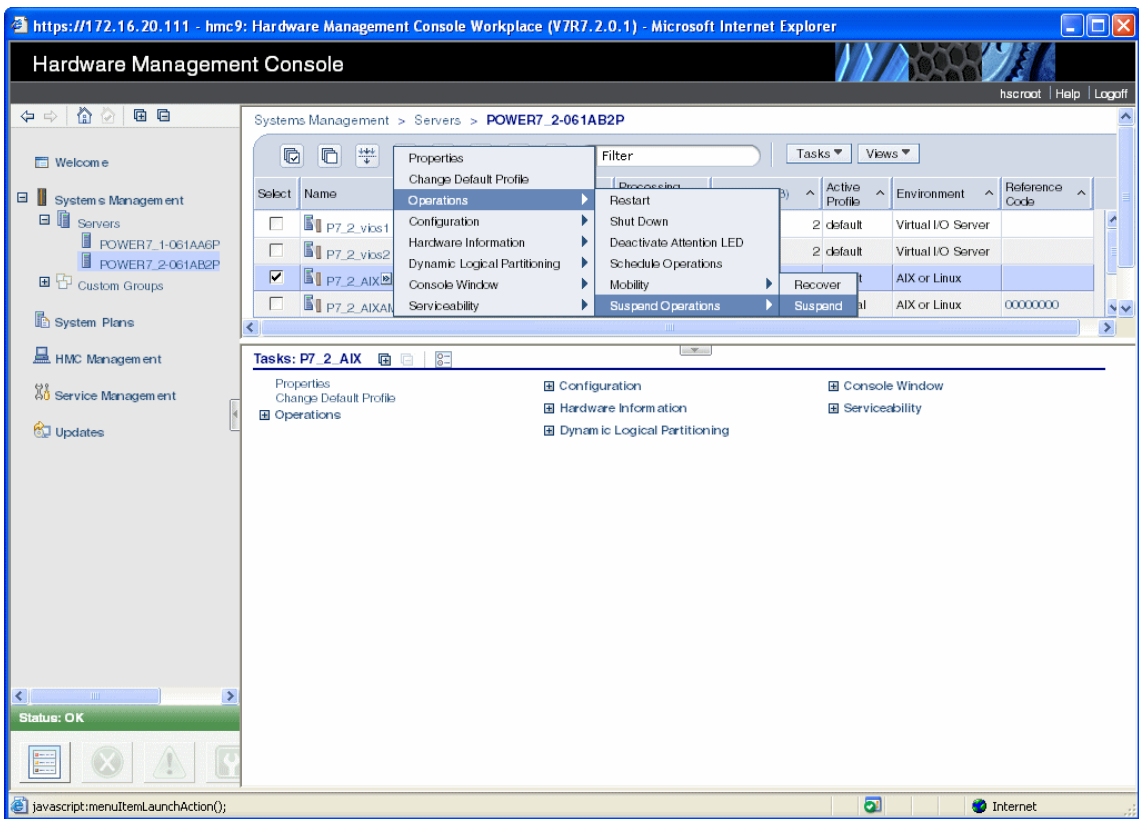


Figure 17-27 Partition suspend menu

2. Verify, and modify if necessary, the information provided in the **Partition Suspend/Resume** window, as shown in Figure 17-28, then click **Validate**.

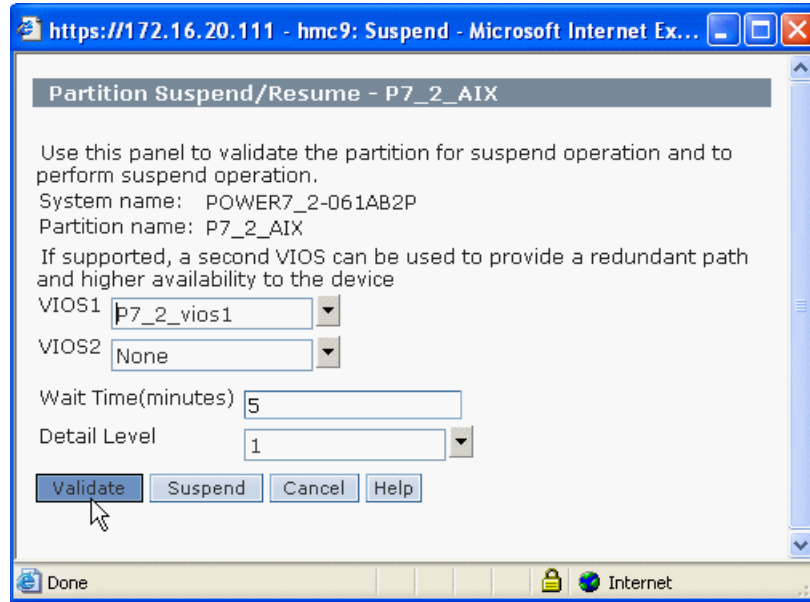


Figure 17-28 Validating suspend operation

3. After completion, the **VALIDATE** windows will provide information regarding the status of the operation, as shown in Figure 17-29. You can now select **OK**.

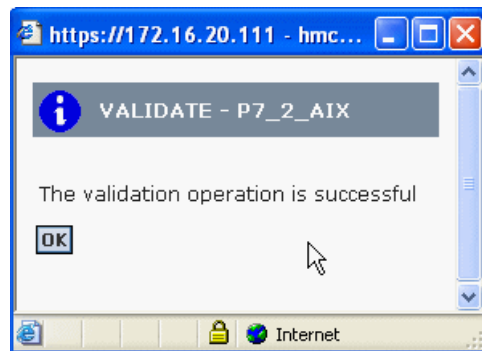


Figure 17-29 Partition successful validation

From the HMC command line interface, use the command **chlparstate** to perform the validation, as shown in Example 17-10.

Example 17-10 Partition suspend validation

```
hscroot@hmc9:~> chlparstate -o validate -t suspend -m POWER7_2-061AB2P  
-p P7_2_AIX -i primary_vios_name=P7_2_vios1  
hscroot@hmc9:~>
```

Requirements: To be resume and suspend capable, a partition needs to meet specific requirements, see Chapter 11.3, “Suspend and Resume planning” on page 297 for partition suspend and resume capability requirements.

17.2.4 Suspending a partition

This section shows how to suspend a partition using the HMC:

1. On the HMC, select the partition you want to suspend, then select **Operations** → **Suspend Operations** → **Suspend** as shown in Figure 17-27 on page 658.
2. Verify, and modify if necessary, the information provided in the **Partition Suspend/Resume** windows, as shown in Figure 17-30, then click **Suspend** when you are ready to perform the suspend operation.

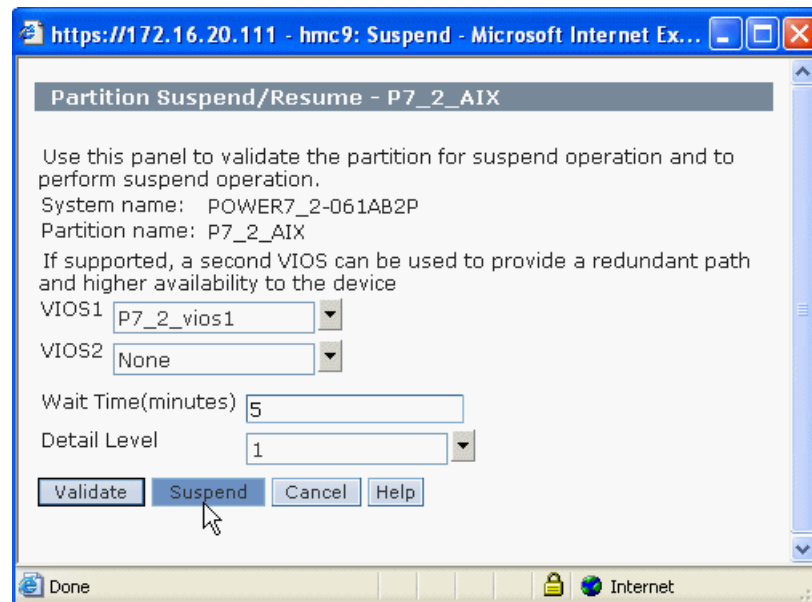


Figure 17-30 Starting partition suspend operation

The suspend operation will start immediately, and a window, as shown in Figure 17-31, will provide information regarding the status and the percentage of completion.

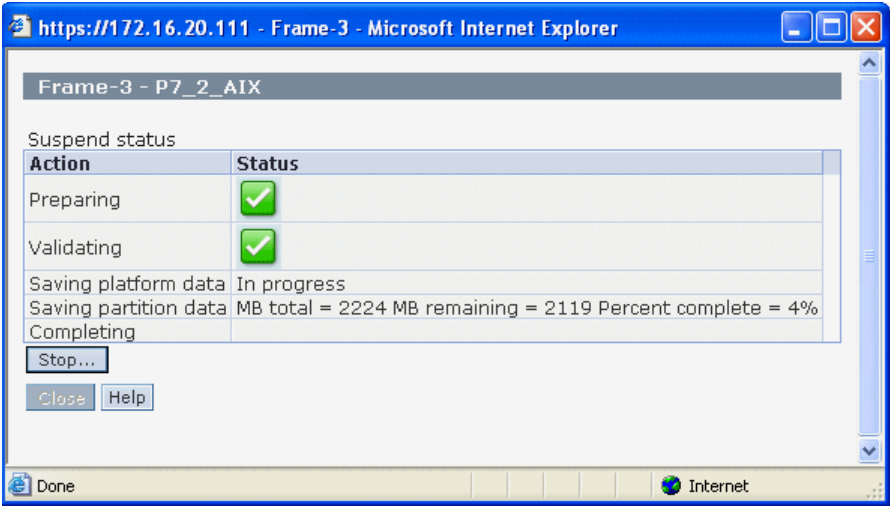


Figure 17-31 Running partition suspend operation

3. After completion, the **Suspend status** windows will provide information regarding the status of the operation, as shown in Figure 17-32. You can now select **Close**.

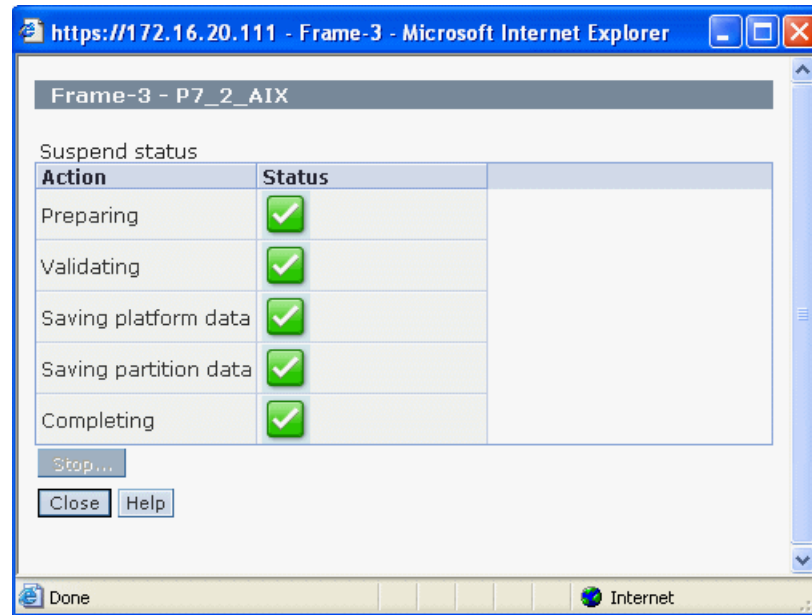


Figure 17-32 Finished partition suspend operation

In the main Hardware Management Console windows, the status of the partition is now **Suspended**, as shown in Figure 17-33.

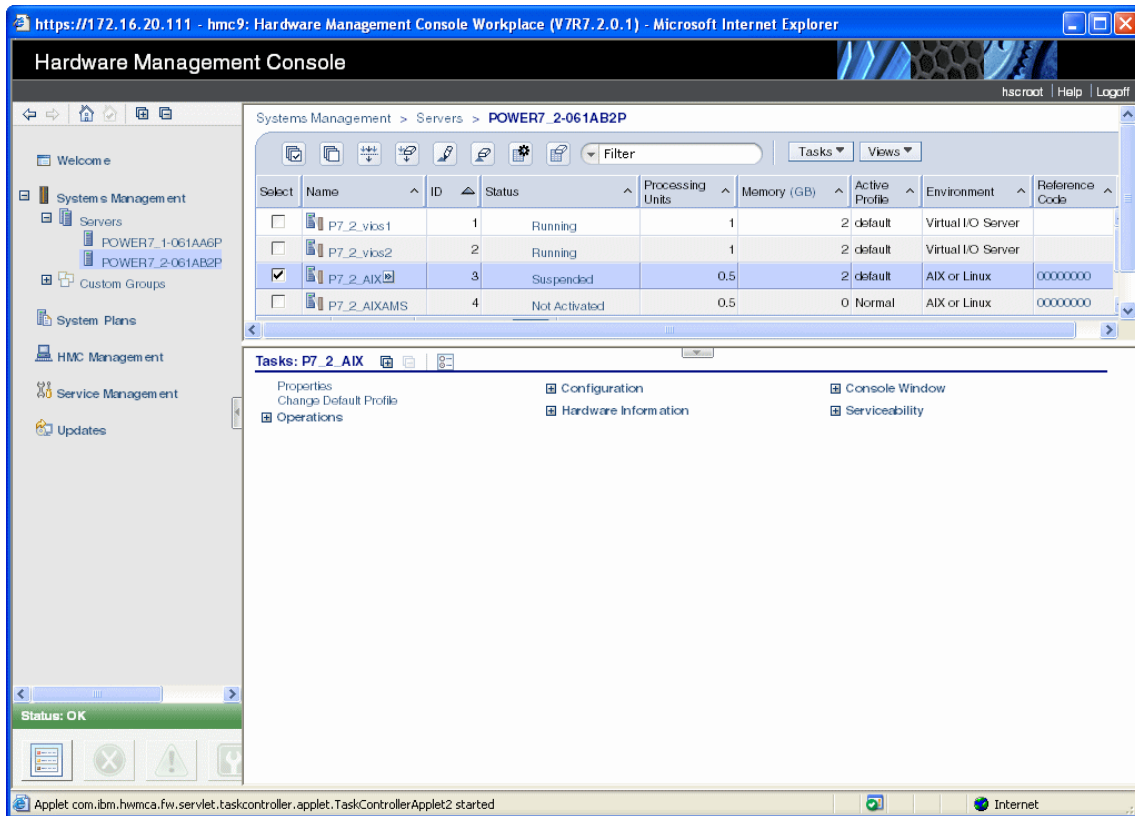


Figure 17-33 Hardware Management Console suspended partition view

Considerations:

- ▶ The suspend operation will remove all virtual devices mapping from the Virtual I/O Servers for the suspended partition.
- ▶ The suspend operation will remove all virtual server adapters used by the suspended partition from the Virtual I/O Servers.
- ▶ The physical volumes used as backing devices by the suspended partition will now appear as volumes that are available for use.

On the HMC, select the suspended partition managed system, then select **Configuration → Virtual Resources → Reserved Storage Device Pool Manager** to view the reserved storage device used to save suspension data for partition as shown in Figure 17-34.

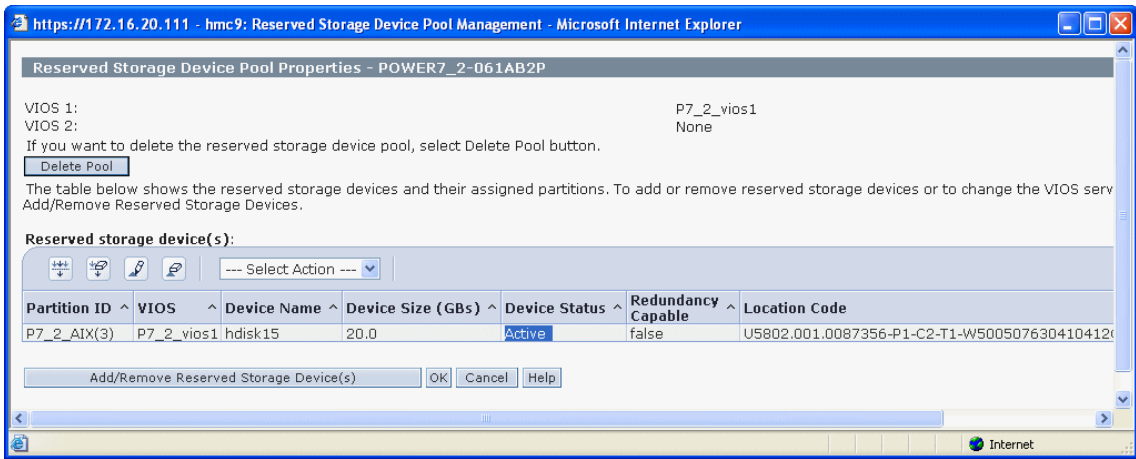


Figure 17-34 Reserved storage device pool properties

From the Hardware Management Console command line interface, the command **lshwres** with the **--rsubtype rsdev** flag can be used view the reserved storage device used to save suspension data for a partition, as shown in Example 17-11.

Example 17-11 Using lshwres command to view reserved storage device pool properties

```
hscroot@hmc9:~> lshwres -r rspool -m POWER7_2-061AB2P --rsubtype rsdev
device_name=hdisk15,vios_name=P7_2_vios1,vios_id=1,size=20480,type=phys
,state=Active,phys_loc=U5802.001.0087356-P1-C2-T1-W500507630410412C-L40
11401200000000,is_redundant=0,lpar_name=P7_2_AIX,lpar_id=3
```

Considerations:

- ▶ Hardware Management Console auto picks an unused and suitable device from the reserved storage device pool to save suspension data.
- ▶ Active Memory Sharing partitions will used their paging space devices to save suspension data.
- ▶ One paging space storage device is required for each partition to be suspended.
- ▶ Storage device required is approximately 110% of the partition's configured maximum memory size.

17.2.5 Validating that a partition is resume capable

The resume validation process verifies that a given partition meets all the requirements to be resumed and that no major configuration issue can prevent this operation from being performed successfully.

As for the suspend validation, the resume validation process is included in the suspend and resume operation wizard.

This section shows how to perform the resume validation operation using the HMC:

1. On the HMC, select the partition you want the suspend validation to be performed on, then select **Operations** → **Suspend Operations** → **Resume** as shown in Figure 17-35.

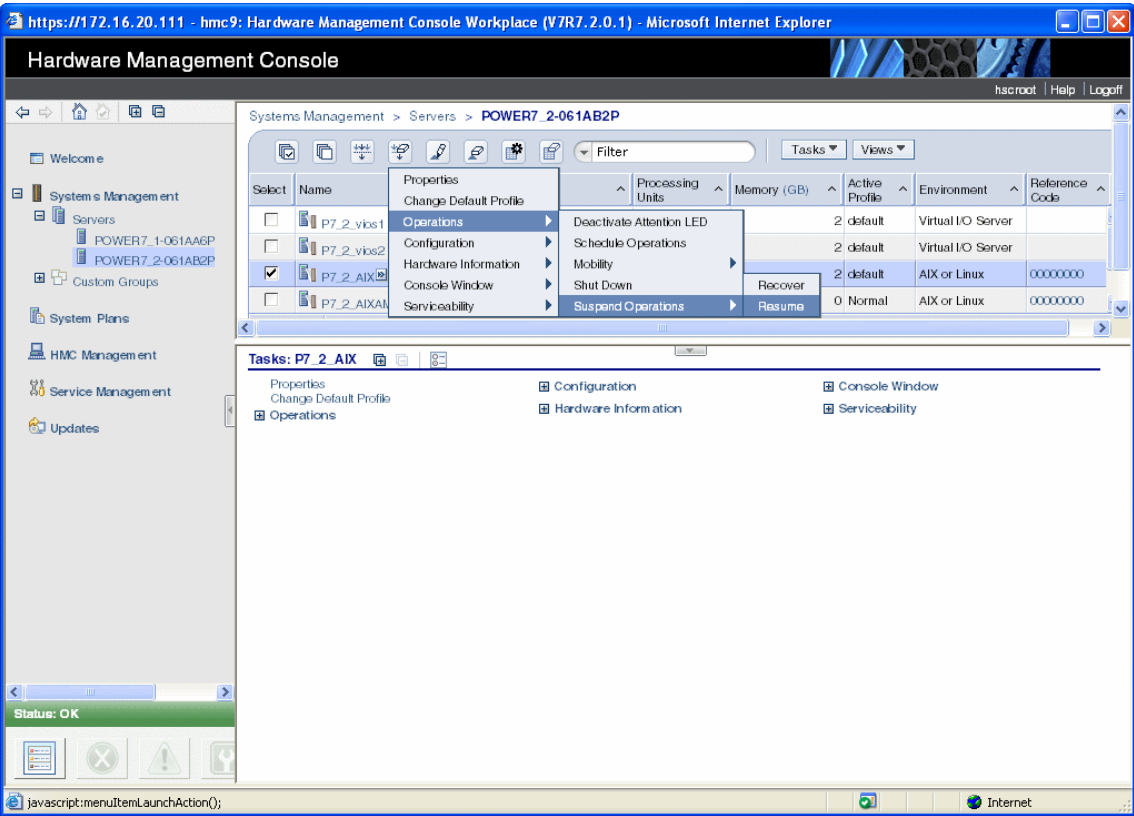


Figure 17-35 Partition resume menu

2. Verify, and modify if necessary, the information provided in the Partition Suspend/Resume window, as shown in Figure 17-36, then click **Validate**.

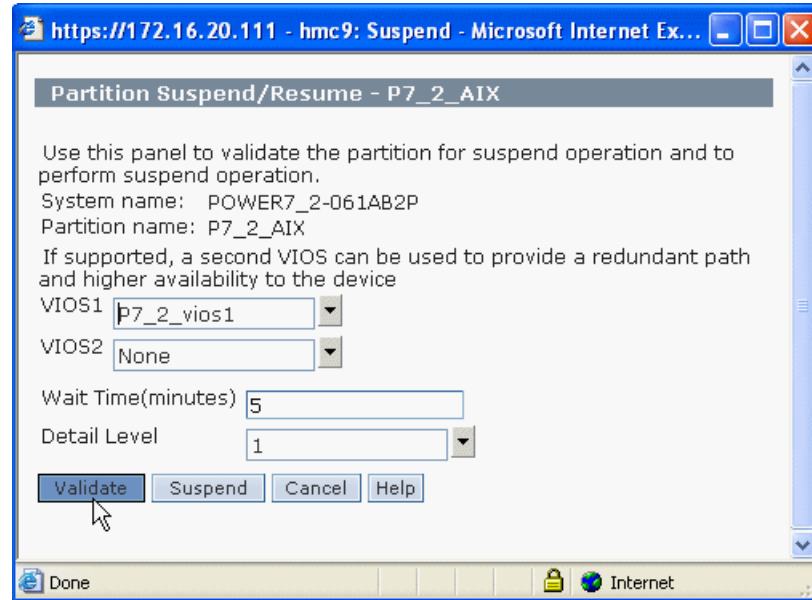


Figure 17-36 Validating resume operation

3. After completion, the VALIDATE window provides information regarding the status of the operation, as shown in Figure 17-37. You can now select **OK**.

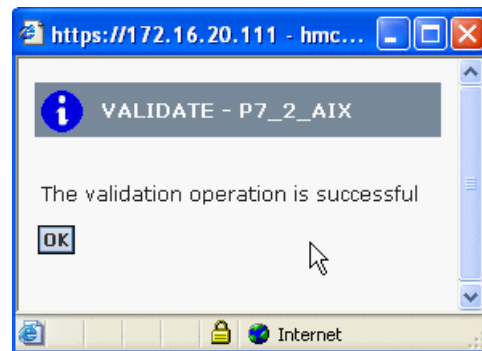


Figure 17-37 Successful validation

From the HMC command line interface, use the command **chlparstate** to perform the validation, as shown in Example 17-12.

Example 17-12 Perform the validation

```
hscroot@hmc9:~> chlparstate -o validate -t resume -m POWER7_2-061AB2P  
-p P7_2_AIX  
hscroot@hmc9:~>
```

17.2.6 Resuming a partition

This section shows how to resume a partition using the HMC:

1. On the HMC, select the suspended partition you want to resume, then select **Operations** → **Suspend Operations** → **Resume** as shown in Figure 17-35 on page 665.
2. Verify, and modify if necessary, the information provided in the Partition Suspend/Resume window, as shown in Figure 17-38, then click **Resume** when you are ready to perform the resume operation.

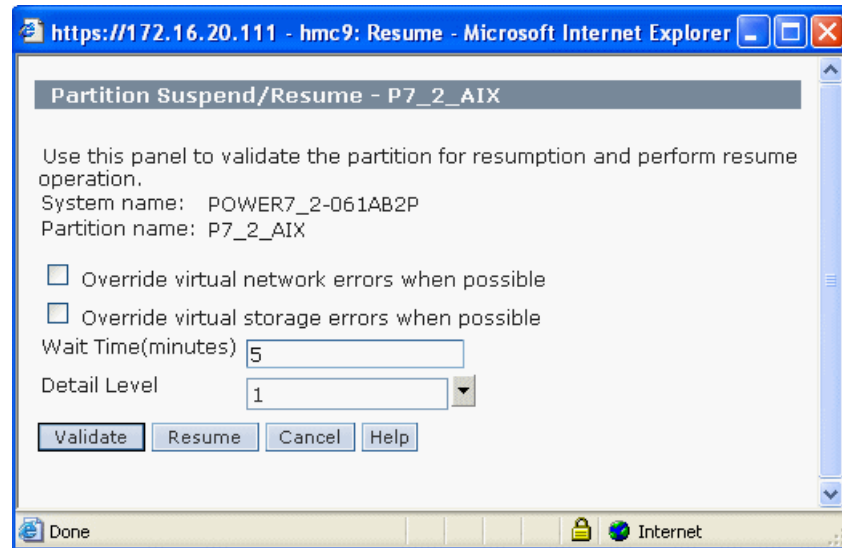


Figure 17-38 Starting partition resume operation

The resume operation starts immediately, and a window, as shown in Figure 17-39, provides information regarding the status and the percentage of completion.

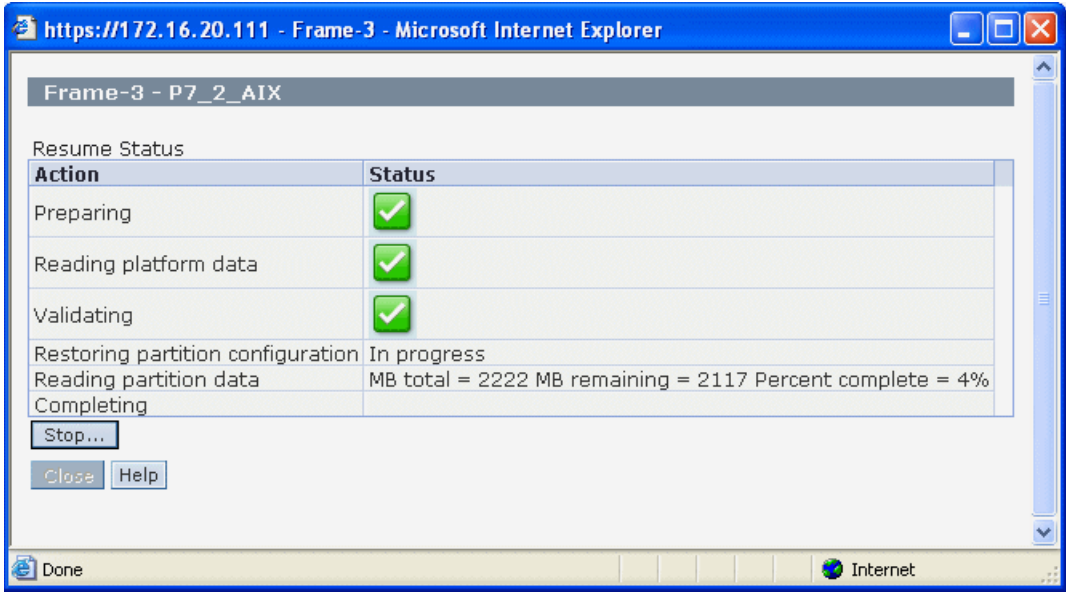


Figure 17-39 Running partition resume operation

3. After completion, the **Resume status** windows provide information regarding the status of the operation, as shown in Figure 17-40. You can now select **Close**.

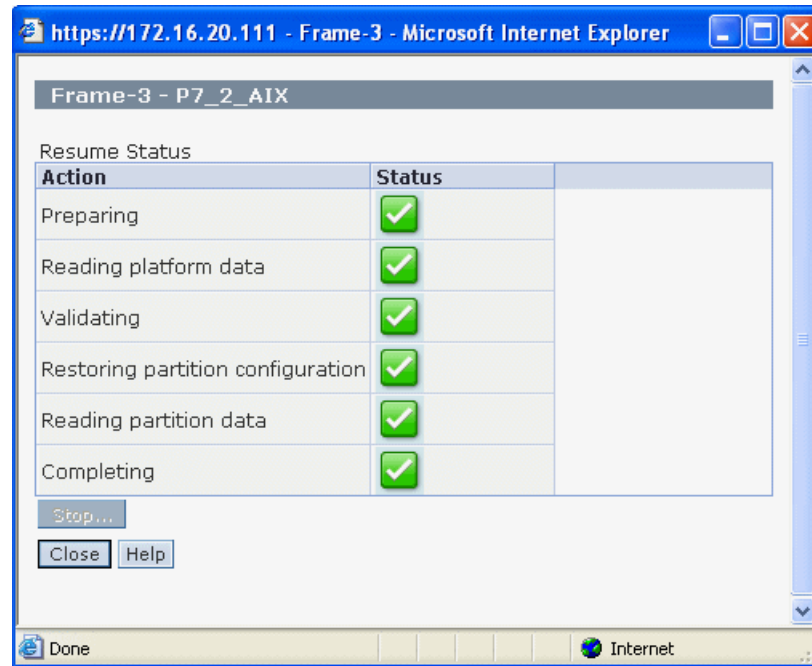


Figure 17-40 Finished partition resume operation

In the main Hardware Management Console windows, the status of the partition is now **Running**, as shown in Figure 17-41.

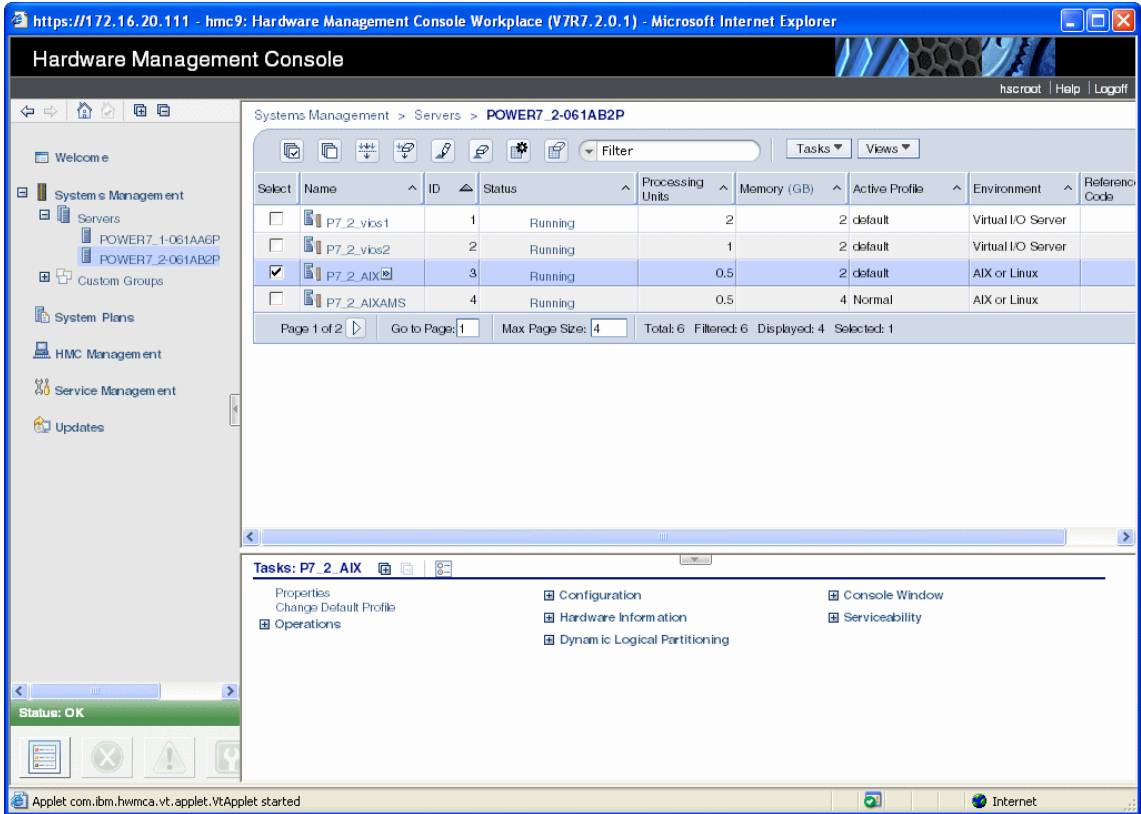


Figure 17-41 Hardware Management Console resume view



Part 5

Appendix

The appendix part of the book comprises the following information:

- ▶ Recent PowerVM enhancements
- ▶ POWER processor modes
- ▶ Capacity on Demand
- ▶ Simultaneous Multithreading
- ▶ Active Memory Expansion
- ▶ IBM i Virtual Partition Manager
- ▶ AIX Workload Partitions
- ▶ System Planning Tool



Recent PowerVM enhancements

Power Systems servers coupled with PowerVM technology are designed to help clients build a dynamic infrastructure, reducing costs, managing risk, and improving service levels.

The following sections provide an overview of new features released for PowerVM and Virtual I/O Server versions.

However, the development of new functions for virtualization is an ongoing process. Therefore, it is best to visit the following website, where you can find more information about the new and existing features:

<http://www14.software.ibm.com/webapp/set2/sas/f/vios/documentation/>

A.1 New features in PowerVM2.2 and Virtual I/O Server Version 2.2 FP26

IBM PowerVM V2.2 and Virtual I/O Server Version 2.2 FP26 and contain the following enhancements:

- ▶ Shared Storage Pools create pools of storage for virtualized workloads, and can improve storage utilization, simplify administration and reduce SAN infrastructure costs. The enhanced capabilities enable 16 systems to participate in a Shared Storage Pool configuration, which can improve efficiency, agility, scalability, flexibility, and availability.
- ▶ Shared Storage Pools flexibility and availability improvements include:
 - IPv6 and VLAN tagging (IEEE 802.1Q) support for intermodal Shared Storage Pools communication
 - Cluster reliability and availability improvements
 - Improved storage utilization statistics and reports
 - Non-disruptive rolling upgrades for applying service
 - Advanced features that accelerate VM deployment, optimize storage utilization, and improve availability through automation.
- ▶ New Virtual I/O Server Performance Advisor analyzes VIOS performance, and makes recommendations for performance optimization.
- ▶ PowerVM has the following new advanced features enabled by VMControl that accelerate VM deployment, optimize storage utilization and improve availability through automation.
 - Linked clones allow for sharing of VM images, which greatly accelerates VM deployment and reduces the storage usage.
 - System pool management for IBM i workloads provides increased flexibility and resource utilization. For further details about the appropriate Systems Director VMControl release, visit:
<http://www.ibm.com/systems/software/director/vmcontrol/>
- ▶ PowerVM will now support 20 VMs per core, which doubles the number of VMs supported per core. This provides additional flexibility, which now allows CPU entitlements as little as 5% of a core.
- ▶ Virtualization support for a dedicated Fibre Channel over Ethernet interface.
- ▶ Usability improvements include the following:
 - Improved Dynamic LPAR console automation, which automatically issues appropriate Virtual I/O Server commands during DLPAR changes initiated from the HMC.
 - Ability for the user to specify the destination Fibre Channel port for any or all virtual Fibre Channel adapters.

- Improved Virtual I/O Server setup, tuning, and validation using the Runtime Expert.
- LPM provides improvements in LPM concurrency and single session mobility performance.

A.2 New features in PowerVM2.2 and Virtual I/O Server Version 2.2 FP25

IBM PowerVM V2.2 and Virtual I/O Server Version 2.2 FP25 contain the following enhancements:

- ▶ Live Partition Mobility provides a 2X improvement in Live Partition Mobility concurrency.
- ▶ Network Load Balancing balances network traffic across redundant Shared Ethernet Adapters.
- ▶ Active Memory Deduplication detects and removes duplicate memory pages to optimize memory usage in Active Memory Sharing configurations.
- ▶ Shared Storage Pools (which requires PowerVM 2.2 Service Pack available December 2011, creates pools of storage for virtualized workloads), and can improve storage utilization, simplify administration, and reduce SAN infrastructure costs. The enhanced capabilities enable four systems to participate in a Shared Storage Pool configuration, which can improve efficiency, agility, scalability, flexibility, and availability.
 - Shared Storage Pools flexibility and availability improvements include:
 - Storage Mobility is a new function that allows data to be moved to new storage devices within Shared Storage Pools while virtual machines remain completely active and available. This ability to move data to new devices can improve performance and allow for the retirement of old storage devices without incurring any application outages.
 - VM Storage Snapshots/Rollback is a new function that allows multiple point-in-time snapshots of individual virtual machine storage. These point-in-time copies can be used to quickly roll back a virtual machine to a particular snapshot image. This functionality can be used to capture a VM image for cloning purposes or before applying maintenance.
 - Shared Storage Pools
 - Increases storage utilization by allowing underutilized storage trapped in traditional storage configurations to be used by a pool of servers. Thinly provisioned virtual disks are an option with Shared Storage Pools, which eliminates the waste of disk resources that are not in use.

In cases where full provisioning of storage is desired up front, thickly provisioned virtual disks are also available.

IBM PowerVM Virtual I/O Server (VIOS) life cycle enhancements

The Virtual I/O Server release and service strategy is being enhanced to provide up to three years of service pack support for each release. Additionally, there will be limited support for some new hardware on previous technology levels. This new release and service policy is limited to future Virtual I/O Server updates starting with Virtual I/O Server V2.2.1 and will not include previous Virtual I/O Server versions.

Improved PowerVM software manageability

IBM intends to provide support for future releases via service pack, and interim fixes for the entire service life of the release. By providing support through service packs and interim fixes throughout the entire service life of a Technology Level, clients would be able to use more consistent service procedures for maintaining Virtual I/O Server.

Support for new hardware on previous releases

Beginning with the Virtual I/O Server V2.2.1 release, service updates will be delivered through both service packs and interim fixes for the entire service life of the Virtual I/O Server release. Enhancements such as processor speed improvements, new I/O adapters, and new processors within a family (for example, IBM POWER7) may be supported on previous releases by installing a service pack

Clients can get support for the new hardware by installing a service pack that contains the necessary changes to support the new hardware on a previous release. This support will typically only include toleration of the new hardware, not exploitation of new hardware features such as additional page sizes.

Exploitation of new hardware features will generally require installing the latest release, although in some cases, upgrading to a later Virtual I/O Server release is necessary to fully exploit new hardware features.

New hardware that requires pervasive changes will not be supported on previous releases. Examples of hardware that would require pervasive changes would include things such as a new processor family or a new I/O bus.

Systems that offer preload or pre-installation of Virtual I/O Server will only include the latest available release. Preload and pre-installation for previous releases of Virtual I/O Server will not be available.

Blade Network Technologies (BNT) VMready Switch

A new IBM BNT® switch feature called IBM VMready® recognizes PowerVM and preserves the IBM BNT switch attributes during Live Partition Mobility events. This new switch feature simplifies the Live Partition Mobility process when BNT VMready switches are deployed in conjunction with IBM Power systems running PowerVM virtual machines.

A.3 New features in Version 2.2 FP24-SP1 of Virtual I/O Server

Virtual I/O Server Version 2.2 FP24-SP1 includes the following enhancements:

- ▶ **Suspend / Resume:**

Using Suspend / Resume, clients can provide long-term suspension (greater than 5-10 seconds) of partitions, saving partition state (memory, NVRAM, and VSP state) on persistent storage, freeing server resources that were in use by that partition, restoring partition state to server resources, and resuming operation of that partition and its applications either on the same server or on a different server.

- ▶ **Requirements for Suspend / Resume:**

All resources must be virtualized prior to suspending a partition. If the partition is to be resumed on a different server, then the shared external I/O (disk and LAN) must remain identical. Suspend / Resume works with AIX and Linux workloads when managed by HMC.

- ▶ **Shared Storage Pools:**

Virtual I/O Server 2.2 allows the creation of storage pools that can be accessed by VIOS partitions deployed across multiple Power Systems servers so that an assigned allocation of storage capacity can be efficiently managed and shared.

- ▶ **Thin provisioning:**

Virtual I/O Server 2.2 supports highly efficient storage provisioning, whereby virtualized workloads in VMs can have storage resources from a Shared Storage Pool dynamically added or released as required.

- ▶ **Virtual I/O Server grouping:**

Multiple Virtual I/O Server 2.2 partitions can utilize a common Shared Storage Pool to more efficiently utilize limited storage resources and simplify the management and integration of storage subsystems.

Virtual I/O Server fix pack 24 provides the following changes, new function, and enhancements:

- ▶ **Role Based Access Control (RBAC):**

RBAC brings an added level of security and flexibility to the administration of Virtual I/O Server. With RBAC, you can create a set of authorizations the user management commands. These authorizations can be assigned to a role “UserManagement,” and then this role can be given to any other user. A normal user with the role “UserManagement” can manage the users on the system but will have no further access. Whenever the system administrator no longer wants to give user management functionalities to “tom”, then he can simply remove the role for user “tom.” With RBAC, Virtual I/O Server will have: the ability to split management functions which presently can be done only by the “padmin” user, provide better security by providing only the necessary access to users, and easy management and auditing of system functions.

- ▶ **USB tape:**

Added support for USB 320 DAT tape drive (Feature Code 5673) and the use of that as a virtual tape device for clients of the Virtual I/O Server.

- ▶ **USB Blu ray:**

USB Blu ray optical devices are now supported. See AIX release notes for further information. The Virtual I/O Server does not support mapping these devices as virtual optical devices to clients. However, you can import the disk into the virtual optical media library, and then map the created file to the client as a virtual DVD drive.

- ▶ **Other enhancements:**

Updates to the IBM Tivoli Monitoring: VIOS Premium Agent and CEC Base Agent version 6.2.2.1.

A.4 New features in Version 2.1 of Virtual I/O Server

IBM PowerVM technology has been enhanced to boost the flexibility of Power Systems servers with support for the following features:

- ▶ **N_Port ID Virtualization (NPIV):**

N_Port ID Virtualization provides direct access to Fibre Channel adapters from multiple client partitions. It simplifies the management of SAN environments. NPIV support is included with PowerVM Express, Standard, and Enterprise Edition.

- ▶ Virtual tape:

Two methods for using SAS tape devices are supported:

 - Access to SAN tape libraries using shared physical HBA resources through NPIV.
 - Virtual tape support allows serial sharing of selected SAS tape devices.
- ▶ Enhancements to PowerVM Live Partition Mobility:

Two major functions have been added to PowerVM Live Partition Mobility:

 - Multiple path support enhances the flexibility and redundancy of Live Partition Mobility.
 - PowerVM Live Partition Mobility is now supported in environments with two Hardware Management Consoles (HMC). This enables larger and flexible configurations.
- ▶ Enhancements to PowerVM Lx86:

PowerVM Lx86 offers the following additional enhancements:

 - Enhanced PowerVM Lx86 installer supports archiving the previously installed environment for backup or migration to other systems.
 - You can automate installation for a non-interactive installation.
 - You can automate installation from an archive.
 - It supports installation using the IBM Installation Toolkit for Linux.
 - SUSE Linux is supported by PowerVM Lx86 when running on RHEL.

For more information about PowerVM Lx86, visit this website:

<http://www.ibm.com/systems/power/software/linux>
- ▶ Enhancements to PowerVM monitoring:

New functions have been added to monitor logical volumes statistics, volume group statistics, network and Shared Ethernet Adapter (SEA) statistics and disk service time metrics. A new **topasrec** tool will help you to record monitoring statistics. A window has also been added to enable you to view Virtual I/O Server and Client throughput.



POWER processor modes

On any Power Systems server, partitions can be configured to run in various modes:

- ▶ **POWER6 compatibility mode:**

This execution mode is compatible with Version 2.05 of the Power Instruction Set Architecture (ISA), which can be found on:

https://power.org/wp-content/uploads/2012/07/PowerISA_V2.05.pdf

- ▶ **IBM POWER6+™ compatibility mode:**

This mode is similar to the POWER6 compatibility mode, with 8 additional Storage Protection Keys.

- ▶ **POWER7 mode:**

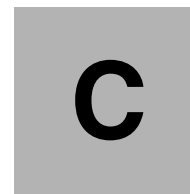
This is the native mode for POWER7 processors, implementing the v2.06 of the Power Instruction Set Architecture,.

https://power.org/wp-content/uploads/2012/07/PowerISA_V2.06B_V2_PUBL IC.pdf

The selection of the mode is made on a per partition basis. Table B-1 lists the differences between these modes.

Table B-1 Differences between POWER6 and POWER7 modes

POWER6 mode (and POWER6+)	POWER7 mode	Customer value
2-thread SMT	4-thread SMT	Throughput performance, processor core utilization
VMX (Vector Multimedia Extension / AltiVec)	VSX (Vector Scalar Extension)	High performance computing
Affinity OFF by default	3-tier memory, micro-partition affinity	Improved system performance for system images spanning sockets and nodes
▶ Barrier synchronization ▶ Fixed 128-byte array; kernel extension access	▶ Enhanced barrier synchronization ▶ Variable size array; user shared memory access	High performance computing parallel programming synchronization facility
▶ 64-core / 128-thread scaling	▶ 32-core / 128-thread scaling ▶ 64-core / 256-thread scaling ▶ 256-core / 1024-thread scaling	Performance and scalability for large scale-up single system image Workloads (such as OLTP, ERP scale-up, WPAR consolidation)
EnergyScale CPU Idle	EnergyScale CPU Idle and Folding with NAP and SLEEP	Improved energy efficiency



Capacity on Demand

Several types of Capacity on Demand (CoD) are available to help on meeting changing resource requirements in an on-demand environment, by using resources that are installed on the system but that are not activated.

The following features are available:

- ▶ Capacity Upgrade on Demand
- ▶ On/Off Capacity on Demand
- ▶ Utility Capacity on Demand
- ▶ Trial Capacity On Demand
- ▶ Capacity Backup
- ▶ Capacity Backup for IBM i
- ▶ MaxCore/TurboCore and Capacity on Demand

Processor resources that are added using Capacity on Demand features are initially added to the default Shared Processor Pool.

Memory resources that are added using Capacity on Demand features are added to available memory on the server. From there they can be added to a Shared Memory Pool or as dedicated memory to a partition.

For software licensing considerations with the various CoD offerings, see the most recent revision of the *Capacity on Demand User's Guide* at this website:

<http://www.ibm.com/systems/power/hardware/cod>



D

Simultaneous Multithreading

Simultaneous Multithreading (SMT) is a method used to increase the throughput for a given amount of hardware. The principle behind SMT is to allow instructions from more than one thread to be executed at the same time on a processor. This allows the processor to continue performing useful work even if one thread has to wait for data to be loaded.

To perform work, a Central Processing Unit needs input information in the form of instructions. Ideally this information will have been loaded into the CPU cache, which allows the information to be quickly accessed. If this information cannot be found in the processor cache, it must be fetched from other storage (other levels of cache, memory or disk) which, in computer terms, can take a long time. While this is happening, the CPU has no information to process, which can result in the CPU idling instead of performing useful work.

The following sections describe SMT technology on IBM POWER Systems.

D.1 POWER processor SMT

The SMT implementation for POWER differs slightly depending on the type of POWER processor. Each new POWER processor generation has been increasing the throughput offered by SMT as seen in most commercial workloads.

Here we compare, at a high level, the implementation of SMT in the different processors:

- ▶ POWER5 uses two separate program counters, one for each thread. Instruction fetches alternate between the two threads. The two threads share the instruction cache.
- ▶ In POWER6, the two threads form a single group of up to seven instructions to be dispatched simultaneously (with up to five from a single thread), increasing the throughput benefit over POWER5 by between 15 to 30 percent.
- ▶ POWER7 enables the execution of four instruction threads simultaneously, offering a significant increase in core efficiency. Additionally it features Intelligent Threads that can vary based on the workload demand. The system either automatically selects (or the system administrator can manually select) whether a workload benefits from dedicating as much capability as possible to a single thread of work, or if the workload benefits more from having capability spread across two or four threads of work. With more threads, the POWER7 processor can deliver more total capacity as more tasks are accomplished in parallel. With fewer threads, those workloads that need very fast individual tasks can get the performance they need for maximum benefit.

Although almost all applications benefit from SMT, some highly optimized workloads might not. For this reason, the POWER processors support *single-threaded* (ST) execution mode. In this mode, the POWER processor gives all the physical processor resources to the active thread.

The benefit of SMT is greatest when there are numerous concurrently executing threads, as is typical in commercial environments, for example, for a Web server or database server. Some specific workloads, for example, certain high performance computing workloads, will generally perform better with ST.

D.2 SMT and the operating system

The operating system scheduler dispatches execution threads to logical processors. Dedicated and virtual processors have one, two or four logical processors associated with them, depending on the number of threads SMT is enabled with. Within each partition it is possible to list the dedicated or virtual processors and their associated logical processors. Because each set of logical processors will be associated with a single virtual or dedicated processor, they will all be in the same partition.

Figure D-1 shows the relationship between physical, virtual, and logical processors. SMT is configured individually for each partition.

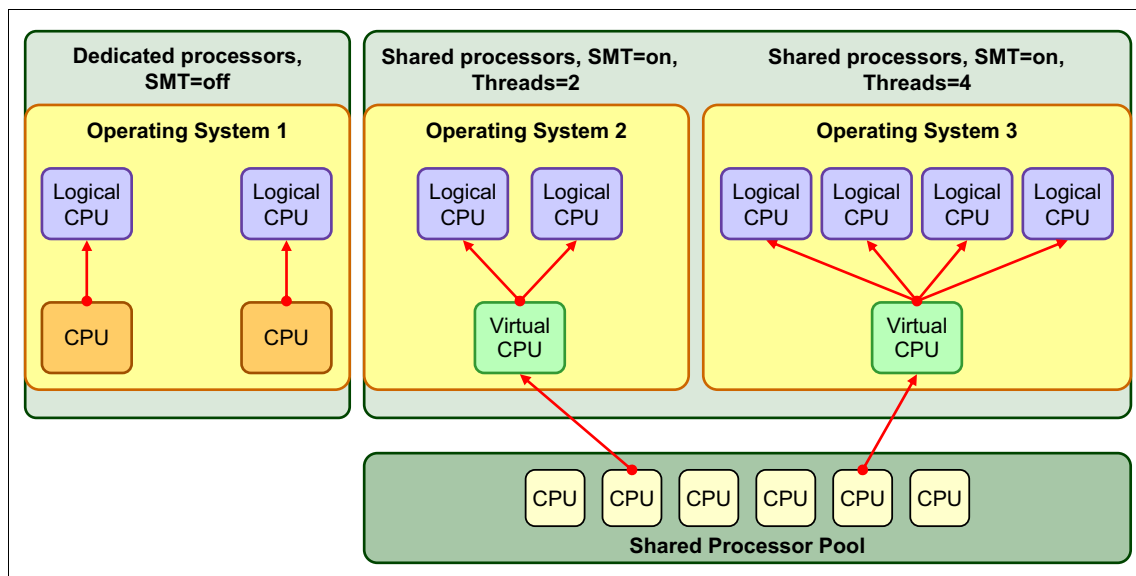


Figure D-1 Physical, virtual, and logical processors

The partition on the left side, hosting operating system 1, has dedicated physical processors assigned and SMT is disabled. For each dedicated physical processor, the operating system sees one logical processor.

The partition in the middle, hosting operating system 2, has SMT enabled with two threads (SMT-2). It is a micro-partition with one virtual processor configured. The operating system sees two logical processors on that virtual processor.

The partition on the right, hosting operating system 3, has SMT enabled with four threads (SMT-4). It is also a micro-partition with one virtual processor. But it sees four logical processors on that virtual processor.

The following sections show how SMT may be controlled on AIX, IBM i, and Linux operating systems.

D.2.1 SMT control in AIX

SMT is controlled by the AIX **smtctl** command or with the System Management Interface Tool (SMIT). SMT can be enabled or disabled in two different ways:

- ▶ Dynamically on a logical partition
- ▶ After the next operating system reboot is performed

D.2.1.1 Setting SMT mode using the command line

The **smtctl** command must be run by users with root authority. The options associated with **smtctl** are **-m**, **-w**, and **-t**; they are defined as follows:

-m off	Will set SMT mode to disabled.
-m on	Will set SMT mode to enabled.
-w boot	Makes the SMT mode change effective on the next and subsequent reboots.
-w now	Makes the mode change effective immediately, but will not persist across reboot.
-t #SMT	Number of threads per processor. On a POWER7 System the valid numbers are 1 (SMT disabled), 2 or 4. If the -t flag is omitted the maximum number of threads are enabled.

SMT: Enabling or disabling SMT can take a while. During the operation the HMC will show a reference code 2000 (when enabling) or 2001 (when disabling).

Rebuilding the boot image

The **smtctl** command does not rebuild the boot image. If you want to change the default SMT mode of AIX, the **bosboot** command must be used to rebuild the boot image. The boot image in AIX Version 5.3 and later has been extended to include an indicator that controls the default SMT mode.

Boots: If neither the **-w boot** nor the **-w now** flags are entered, the mode change is made immediately and will persist across reboots. The boot image must be remade with the **bosboot** command in order for a mode change to persist across subsequent boots, regardless of **-w** flag usage.

The **smtctl** command entered without a flag will show the current state of SMT in the partition. Example D-1 shows an example where SMT is enabled using the **smtctl** command.

Example D-1 Enabling SMT using the smtctl command

```
# smtctl -m on
smtctl: SMT is now enabled. It will persist across reboots if
        you run the bosboot command before the next reboot.
# smtctl
```

```
This system is SMT capable.
This system supports up to 4 SMT threads per processor.
SMT is currently enabled.
SMT boot mode is set to enabled.
SMT threads are bound to the same virtual processor.
```

```
proc0 has 4 SMT threads.
Bind processor 0 is bound with proc0
Bind processor 2 is bound with proc0
Bind processor 3 is bound with proc0
Bind processor 4 is bound with proc0
```

```
proc4 has 4 SMT threads.
Bind processor 1 is bound with proc4
Bind processor 5 is bound with proc4
Bind processor 6 is bound with proc4
Bind processor 7 is bound with proc4
```

D.2.1.2 Setting SMT mode using SMIT

Use the **smitty smt** fast path to access the SMIT SMT control panel. From the main SMIT panel, the selection sequence is **Performance & Resource Scheduling Simultaneous Multi-Threading Processor Mode Change SMT Mode**. Figure D-2 shows the SMIT SMT panel.

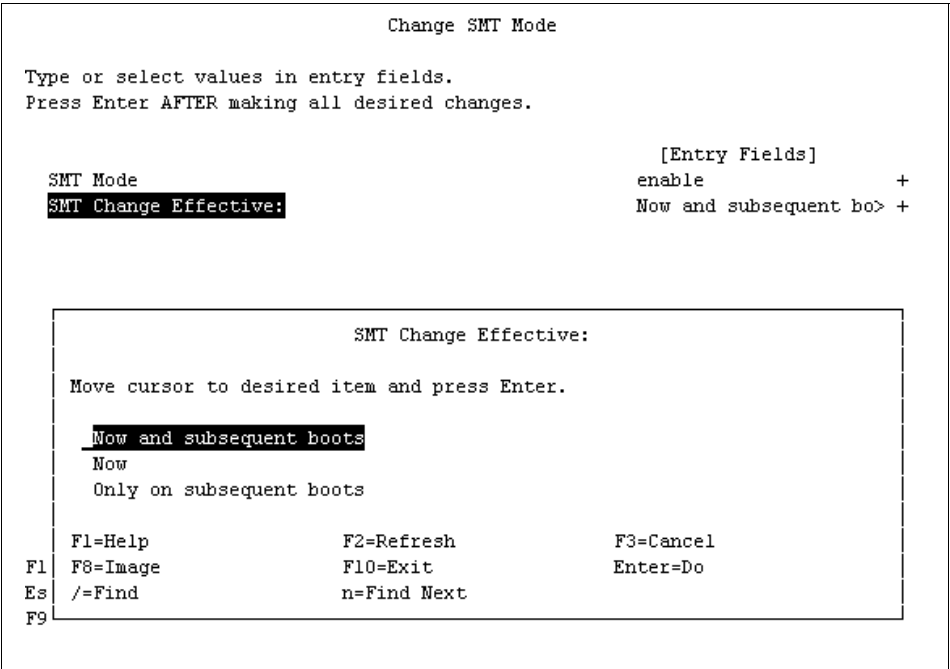


Figure D-2 SMIT SMT panel with options

SMT: It is not possible to specify the number of threads when enabling SMT using SMIT.

D.2.1.3 SMT performance monitor and tuning

AIX includes additional commands or extended options to existing commands for the monitoring and tuning of system parameters in SMT mode. For more information, see *IBM PowerVM Virtualization Managing and Monitoring*, SG24-7590.

D.2.2 SMT control in IBM i

Usage of the SMT processor feature by IBM i is controlled by the QPRCMLTTSK system value. A change of this system value requires an IPL of the IBM i partition to make it effective. Because IBM i automatically makes use of an existing SMT processor capability with the default QPRCMLTTSK=2 setting as shown in Figure D-3, this system value typically must not be changed.

```

                                Display System Value

System value . . . . . :  QPRCMLTTSK
Description . . . . . :  Processor multi tasking

Processor multi tasking . :  2          0=Off
                                1=On
                                2=System-controlled

.....
:          Processor multitasking - Help          :
:                                                  :
:  0          Processor multitasking is turned off  :
:                                                  :
:  1          Processor multitasking is turned on   :
:                                                  :
:  2          Processor multitasking is set to      :
:          System-controlled                        :
:                                                  :
:                                                  : Bottom :
Press Enter t :  F2=Extended help  F3=Exit help    F10=Move to top :
               :  F12=Cancel    F13=Information Assistant  F14=Print help :
F3=Exit  F12 :
:.....

```

Figure D-3 IBM i processor multi-tasking system value

D.2.3 SMT control in Linux

SMT can be enabled or disabled at boot time or dynamically after the partition is running.

D.2.3.1 Controlling SMT at boot time

After the next operating system reboot is performed, to enable or disable SMT at boot, use the following boot option at the boot prompt:

```
boot: linux smt-enabled=on
```

Change the **on** to **off** to disable SMT at boot time. The default is SMT on. On a POWER5 or POWER6 based server, SMT-2 will be enabled. On a POWER7 server with a POWER7 enabled kernel, SMT-4 will be enabled.

D.2.3.2 Controlling SMT using the `ppc64_cpu` command

When Linux is up and running, SMT can be controlled using the `ppc64_cpu` command. Example D-2 shows how SMT can be turned on dynamically. After SMT has been enabled, Linux sees eight processors in `/proc/cpuinfo`.

Example D-2 Using `ppc64_cpu` to control SMT on Linux

```
[root@localhost ~]# cat /proc/cpuinfo
processor: 0
cpu      : POWER7 (architected), altivec supported
clock    : 3550.000000MHz
revision : 2.3 (pvr 003f 0203)

processor: 1
cpu      : POWER7 (architected), altivec supported
clock    : 3550.000000MHz
revision : 2.3 (pvr 003f 0203)

processor: 2
cpu      : POWER7 (architected), altivec supported
clock    : 3550.000000MHz
revision : 2.3 (pvr 003f 0203)

processor: 3
cpu      : POWER7 (architected), altivec supported
clock    : 3550.000000MHz
revision : 2.3 (pvr 003f 0203)

processor: 4
cpu      : POWER7 (architected), altivec supported
```

clock : 3550.000000MHz
revision: 2.3 (pvr 003f 0203)

processor: 5
cpu : POWER7 (architected), altivec supported
clock : 3550.000000MHz
revision: 2.3 (pvr 003f 0203)

processor: 6
cpu : POWER7 (architected), altivec supported
clock : 3550.000000MHz
revision: 2.3 (pvr 003f 0203)

processor: 7
cpu : POWER7 (architected), altivec supported
clock : 3550.000000MHz
revision: 2.3 (pvr 003f 0203)

timebase: 512000000
platform: pSeries
model : IBM,8205-E6C
machine: CHRP IBM,8205-E6C

**E**

Active Memory Expansion

Active Memory Expansion is an innovative POWER7 (or later) technology that allows the effective maximum memory capacity to be much larger than the true physical memory maximum. Compression and decompression of memory content can allow memory expansion up to 100%. This can allow a partition to do significantly more work or support more users with the same physical amount of memory. Similarly, it can allow a server to run more partitions and do more work for the same physical amount of memory.

Feature: Active Memory Expansion is not a PowerVM capability but has to be ordered as separate feature code #4971 or feature code #4792, depending on the server type and model.

E.1 Prerequisites

Active Memory Expansion requires the following:

- ▶ POWER7 (or later) System with Active Memory Expansion feature enabled
- ▶ HMC V7R7.1.0 or later
- ▶ AIX 6.1 Technology Level 6 or later

E.2 Overview

When configuring a partition with Active Memory Expansion, the following two settings define how much memory will be available:

Physical memory	This is the amount of physical memory available to the partition. Usually this corresponds to the desired memory in the partition profile.
Memory expansion factor	Defines how much of the physical memory will be expanded.

Tip: The memory expansion factor can be defined individually for each partition.

The amount of memory available to the operating system can be calculated by multiplying the physical memory with the memory expansion factor.

For example, in a partition that has 10 GB of physical memory and is configured with a memory expansion factor of 1.5, the operating system will see 15 GB of available memory.

Figure E-1 shows an example partition that has Active Memory Expansion enabled.

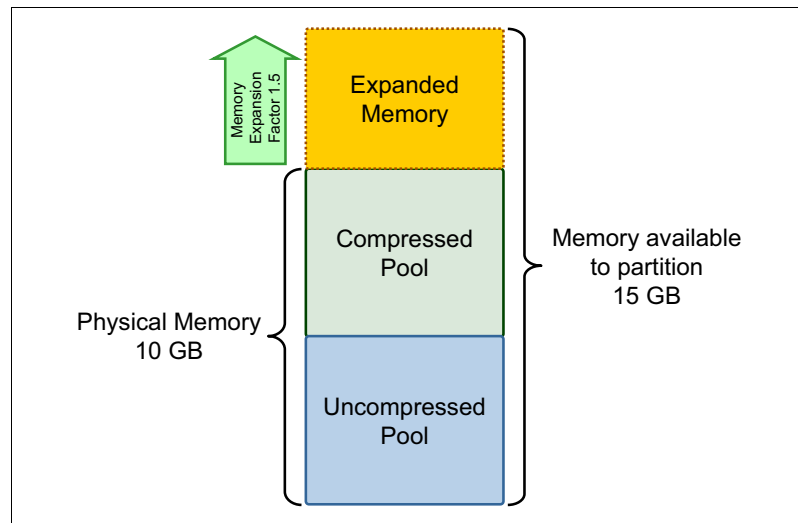


Figure E-1 Active Memory Expansion example partition

The partition has 10 GB of physical memory assigned. It is configured with a memory expansion factor of 1.5. This results in 15 GB of memory that is available to the operating system running in the partition. The physical memory is separated into the following two pools:

Uncompressed pool Contains the non-compressed memory pages that are available to the operating system just like normal physical memory

Compressed pool Contains the memory pages that have been compressed by Active Memory Expansion

Some parts of the partition memory are located in the uncompressed pool, others in the compressed pool. The size of the compressed pool changes dynamically.

Depending on the memory requirements of the application, memory is moved between the uncompressed and compressed pool.

When the uncompressed pool gets full, Active Memory Expansion will compress pages that are infrequently used and move them to the compressed pool to free up memory in the uncompressed pool.

When the application references a compressed page, Active Memory Expansion will decompress it and move it to the uncompressed pool.

The pools, and also the compression and decompression activities that take place when moving pages between the two pools, are transparent to the application.

The compression and decompression activities require CPU cycles. Therefore, when enabling Active Memory Expansion, there have to be spare CPU resources available in the partition for Active Memory Expansion.

Active Memory Expansion does not compress file cache pages and pinned memory pages.

If the expansion factor is too high, the target expanded memory size cannot be achieved and a memory deficit forms. The effect of a memory deficit is the same as the effect of configuring a partition with too little memory. When a memory deficit occurs, the operating system might have to resort to paging out virtual memory to the paging space.

E.3 Tools

AIX provides the **amepat** command, which can be used to analyze existing workloads. The **amepat** command shows statistics on the memory usage of a partition and provides suggestions for Active Memory Expansion configurations, including the estimated CPU usage.

The **amepat** command can be run on any system supported by AIX 6.1 or later. It can therefore be run on older systems such as a POWER4 based system before consolidating the workload to an Active Memory Expansion enabled POWER7 based system.

The **amepat** command can also be used to monitor the performance. For more details, see *IBM PowerVM Virtualization Managing and Monitoring*, SG24-7590 available at:

<http://www.redbooks.ibm.com/abstracts/sg247590.html>

E.4 Active Memory Expansion setup

The following example shows how to enable Active Memory Expansion for an existing AIX partition. The partition used in this example initially has 10 GB of physical memory assigned.

We assume that on the server where the partition is running, another partition needs more physical memory. Because no spare memory is available, the memory footprint of the example partition has to be reduced.

As a first step, the **amepat** command is run to analyze the workload in the partition and get a suggestion for a reasonable physical memory size and memory expansion factor. Example E-1 shows the **amepat** command output.

Example E-1 amepat command example

```
.
.[Lines omitted for clarity]
.
Active Memory Expansion Modeled Statistics      :
-----
Modeled Expanded Memory Size : 10.00 GB
Achievable Compression ratio :2.85

Expansion      Modeled True      Modeled      CPU Usage
Factor          Memory Size      Memory Gain      Estimate
-----
      1.03          9.75 GB      256.00 MB [ 3%]  0.00 [ 0%]
      1.22          8.25 GB       1.75 GB [ 21%]  0.00 [ 0%]
      1.38          7.25 GB       2.75 GB [ 38%]  0.00 [ 0%]
      1.54          6.50 GB       3.50 GB [ 54%]  0.00 [ 0%]
      1.67          6.00 GB       4.00 GB [ 67%]  0.00 [ 0%]
      1.82          5.50 GB       4.50 GB [ 82%] 0.00 [ 0%]
      2.00          5.00 GB       5.00 GB [100%]  0.52 [ 26%]
.
.[Lines omitted for clarity]
.
```

In this case the optimum memory size is 5.5 GB with a memory expansion factor of 1.82. With these settings, the operating system in the partition will still see 10 GB of available memory, but the amount of physical memory can be reduced by almost half. A higher expansion factor means that significantly more CPU resources will be needed for performing the compression and decompression.

Figure E-2 shows the required updates in the partition profile to achieve the memory savings just described. First the amount of physical memory is reduced from 10 GB to 5.5. GB. To enable Active Memory Expansion the check box in the Active Memory Expansion section of the Memory tab at the bottom has to be checked and the memory expansion factor must be set. In our example a memory expansion factor of 1.82 is defined.

https://172.16.20.111 - hmc9: Manage Profiles - Microsoft Internet Explorer

Logical Partition Profile Properties: default @ P7_1_AIX @ POWER7_1-061AA6P - P7_1_AIX

General Processors **Memory** I/O Virtual Adapters Power Controlling Settings HCA Logical Host Ethernet Adapters (LHEA)

Detailed below are the current memory settings for this partition profile.

Dedicated Memory
 Installed memory (MB): 131072
 Current memory available for partition usage (MB): 129536

Minimum memory : 0 GB 256 MB
 Desired memory : 5 GB 512 MB
 Maximum memory : 20 GB 0 MB

Specify the Barrier Synchronization Register **BSR** for this profile

Available BSR arrays: 0
 BSR arrays for this profile: 0

Huge Page Memory
 Page size (in GB) : 16
 Configurable pages : 0
 Minimum pages : 0
 Desired pages : 0
 Maximum pages : 0

Active Memory Expansion
☒ Active memory expansion factor (1.00 - 10.00) 1.82

OK Cancel Help

Figure E-2 Enabling Active Memory Expansion on the HMC

To enable Active Memory Expansion the partition has to be deactivated and reactivated.

As you can see in Example E-2, the partition shows 10 GB of memory available after the reboot. When displaying the detailed partition configuration using the **lparstat -i** command, the amount of physical memory and the expansion factor are displayed.

Example E-2 Using lparstat to display AME configuration

```
# lparstat

System configuration: type=Shared mode=Uncapped smt=4 lcpu=8
mem=10240MB psize=14 ent=0.20

%user  %sys  %wait  %idle  physc  %entc  lbusy  vcsw  phint
-----
  0.0   0.0   0.0   100.0   0.00   0.0    0.5  239527    8

# lparstat -i
Node Name                : P7_1_AIX
Partition Name           : P7_1_AIX
Partition Number         : 3
Type                     : Shared-SMT-4
Mode                     : Uncapped
Entitled Capacity        : 0.20
Partition Group-ID       : 32771
Shared Pool ID           : 0
Online Virtual CPUs      : 2
Maximum Virtual CPUs     : 4
Minimum Virtual CPUs     : 1
Online Memory           : 5632 MB
Maximum Memory           : 20480 MB
Minimum Memory           : 256 MB
Variable Capacity Weight : 128
Minimum Capacity         : 0.10
Maximum Capacity         : 2.00
Capacity Increment       : 0.01
Maximum Physical CPUs in system : 16
Active Physical CPUs in system : 16
Active CPUs in Pool      : 14
Shared Physical CPUs in system : 14
Maximum Capacity of Pool : 1400
Entitled Capacity of Pool : 20
Unallocated Capacity     : 0.00
Physical CPU Percentage  : 10.00%
Unallocated Weight       : 0
```

Memory Mode	: Dedicated-Expanded
Total I/O Memory Entitlement	: -
Variable Memory Capacity Weight	: -
Memory Pool ID	: -
Physical Memory in the Pool	: -
Hypervisor Page Size	: -
Unallocated Variable Memory Capacity Weight	: -
Unallocated I/O Memory entitlement	: -
Memory Group ID of LPAR	: -
Desired Virtual CPUs	: 2
Desired Memory	: 5632 MB
Desired Variable Capacity Weight	: 128
Desired Capacity	: 0.20
Target Memory Expansion Factor	: 1.82
Target Memory Expansion Size	: 10240 MB
Power Saving Mode	: Disabled

#

Tip: After Active Memory Expansion has been enabled, the memory expansion factor can be changed dynamically using DLPAR.



IBM i Virtual Partition Manager

The Virtual Partition Manager (VPM) is a feature of i5OS that enables the creation and management of logical partitions and does not require the use of Hardware Management Console (HMC) or Integrated Virtualization Manager (IVM)

The following sections discusses:

- ▶ Planning considerations
- ▶ Creating an IBM i client partition using VPM
- ▶ Creating a Linux partition using VPM

F.1 Planning considerations

It is strongly recommended that you fully understand the planning considerations required to enable Virtual Partition Manager.

- ▶ Virtual Partition Manager support is provided through either Dedicated Service Tools (DST) or System Service Tools (SST).
- ▶ Beginning with IBM i V7R1, the Virtual Manager Partition has been enhanced and allows up to four IBM i partitions to be enabled.
- ▶ A maximum of four Linux partitions are supported.
- ▶ I/O for all client partitions must be managed by a single IBM i server partition.
- ▶ Up to a maximum of four virtual Ethernet connections may be configured for each Linux partition or for the IBM i partition.
- ▶ Automatic processor balancing between Linux and IBM i partitions is supported through uncapped processor pools.
- ▶ Dynamic movement of resources (DLPAR) such as processor, memory, and I/O resources is not supported. When resources are changed in an IBM i or Linux partition, an IPL is required.
- ▶ Starting with IBM i V7R1, the Virtual Partition Manager now supports Ethernet layer-2 bridging between a physical network and the Power Systems virtual Ethernet. Using layer-2 bridging, one Ethernet port in an IBM i partition can provide network access to other logical partitions on the same platform. This is similar functionality to the Shared Ethernet Adapter (SEA) support provided by the Power Systems Virtual I/O Server (VIOS) partition.
- ▶ Partition configuration data cannot be saved through DST or SST tasks. Ensure that hardcopy prints are kept with configuration screens should you need to recreate the partitions.
- ▶ The IBM i Client partitions can either be IBM i 7.1, or IBM i 6.1 on POWER6 Systems or later.
- ▶ You cannot use Virtual Partition Manager on an IBM i server that is configured using an HMC.

F.2 Creating an IBM i client partition using VPM

Creating an IBM i Client Partition using Virtual Partition Manager is described in detail in the following publication *Creating IBM i Client Partitions Using Virtual Partition Manager*, REDP-4806.

F.3 Creating a Linux partition using VPM

Creating Linux Partition using Virtual Partition Manager is described in detail in the following publication *Virtual Partition Manager A Guide to Planning and Implementation*, REDP-4013.



AIX Workload Partitions

With the release of AIX 6.1, IBM introduced a new virtualization capability called Workload Partition (WPAR). A WPAR is a software-created, virtualized operating system environment within a single AIX image. Each WPAR is a secure and isolated environment for the application that it hosts. The application in a WPAR “thinks” that it is being executed in its own dedicated AIX instance.

WPARs can be created in all hardware environments that support AIX 6.1 and later. This includes, for example, POWER4 machines, which are supported by AIX 6.1.

Figure G-1 shows that you can create WPARs within multiple AIX instances on the same physical server, whether they execute in dedicated LPARs or micro-partitions.

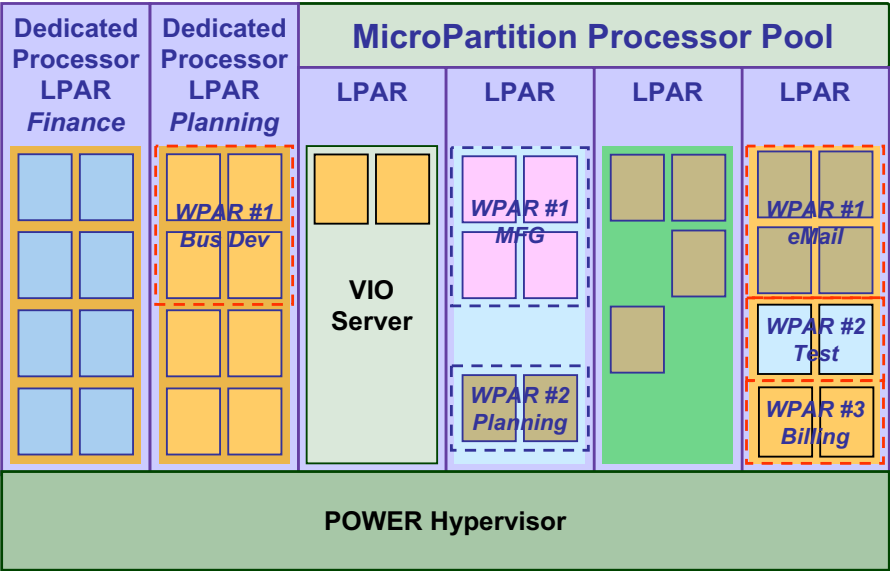


Figure G-1 WPAR instantiated within dedicated partitions and micro-partitions

G.1 Characteristics of WPARs

Workload partitions have the following characteristics:

- ▶ The workload partition technology can help in an environment where an application environment needs to be started often, on-demand, and quickly, which might apply, for example, to test environments.
- ▶ You can use the configuration that is stored in specification files as input to WPAR creation commands, which allows the system administrator to automate, through scripts and programs, the startup and handling of multiple workload partitions.
- ▶ The WPAR technology gives you additional flexibility in system capacity planning as part of a strategy for maximizing system utilization and provisioning efficiency.
- ▶ AIX 6.1 and later provides highly granulated control of processor and memory resource allocation to workload partitions (down to 0.01% increments). This technology is therefore suitable for server consolidation of very small workloads.
- ▶ The WPAR technology allows you to share an AIX instance between multiple applications, while still running each application within its own environment, which provides isolation between applications. In this case, the more applications that are consolidated within one AIX instance, the less the system administrator has to perform operating system (OS) fixes, backups, migration, and other maintenance tasks.

G.2 Types of WPARs

There are multiple types of workload partitions:

- ▶ System WPARs provide an entire virtualized operating system environment.
- ▶ Application WPARs provide isolation for individual services or applications.

Both types can run within a single AIX OS image, which is referred to as the global environment. Starting with AIX 7.1 and POWER7, versioned WPARs are available. Versioned WPARs provide a different OS version in the WPAR than the OS version of the global environment, for example, an AIX 5.2 WPAR running in an AIX 7.1.

G.2.1 System WPARs

A System WPAR presents a secure and isolated environment that is most similar to a standalone AIX system. Each System WPAR has dedicated writable file systems, although it can share the global environment `/usr` and `/opt` file systems in read only mode. Here, we term this difference as Unshared and Shared, respectively.

System WPARs are autonomous virtual system environments and appear, to applications that are running within the WPAR, as though they run in their own separate instance of AIX. Multiple System WPARs can run within the single global AIX image, and each WPAR is isolated from other WPARs.

System WPARs have the following attributes:

- ▶ All typical system services are activated.
- ▶ Operating system services, such as telnet, are supported. You can telnet into a system WPAR as root.
- ▶ Use distinct writeable file systems that are not shared with any other WPAR or the global system.
- ▶ Own private file systems.
- ▶ Own set of users, groups, and network resources.
- ▶ WPAR root user has no authority in the global environment.

The combination of isolation and resource sharing makes System WPARs an excellent feature for consolidating several systems, independently from the underlying POWER hardware. Using a single global AIX image simplifies OS maintenance for multiple systems; however, using one global OS image also introduces a very obvious single-point-of-failure and introduces new dependencies.

G.2.2 Application WPARs

An Application WPAR has all of the process isolation that the System WPAR does. However, it shares the file system name space with the global system and any other Application WPAR that is defined within the system.

An Application WPAR is essentially a wrapper around a running application or process for the purposes of isolation and mobility. It lacks some of the system services that the System WPAR provides. For example, it is not possible to log in or to telnet into an Application WPAR. When the application that is running in an Application WPAR terminates, the WPAR also ceases to exist.

For more information about WPARs, refer to *Exploiting IBM AIX Workload Partitions*, SG24-7955, available at:

<http://www.redbooks.ibm.com/redbooks/pdfs/sg247955.pdf>

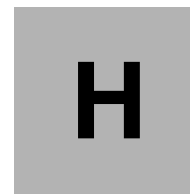
G.3 Live Application Mobility

Live Application Mobility requires an additional Licensed Program Product called IBM Workload Partitions Manager™ for AIX.

The Workload Partitions Manager provides more advanced features for managing multiple WPARs, including these:

- ▶ Graphical interface with wizards for many management tasks
- ▶ Resource allocation
- ▶ Role-based views and tasks
- ▶ Centralized, single point-of-administration
- ▶ Automated application mobility based on policies

To transfer an application to another LPAR or server, a checkpoint/restart feature is used. During a WPAR checkpoint, the current state of running applications is saved and during restart operations resumed in the new AIX image. With this feature, it is possible to move WPARs between compatible systems without significantly affecting WPAR users.



System Planning Tool

This appendix describes how you can use the PC-based System Planning Tool (SPT) to create a configuration to be deployed on a system. When deploying the partition profiles, assigned resources are generated on the Hardware Management Console (HMC) or Integrated Virtualization Manager (IVM). The Virtual I/O Server operating system can be installed during the deployment process. In the scenario described in this chapter, the Virtual I/O Server, AIX, IBM i, and Linux operating systems are all installed using DVD or NIM.

SPT is available as a download from the System Planning Tool website for no additional charge. The generated system plan can be viewed from the SPT on a PC, or directly on an HMC. After you save your changes, the configuration can be deployed to the HMC or IVM. For detailed information about the SPT, see the following address:

<http://www.ibm.com/systems/support/tools/systemplanningtool>

The next sections present the following topics:

- ▶ Sample scenario
- ▶ Preparation recommendations
- ▶ Planning the configuration with SPT
- ▶ Initial setup checklist

H.1 Sample scenario

This scenario shows the system configuration used for this book.

Figure H-1 shows the basic layout of partitions and the slot numbering of virtual adapters. An additional virtual SCSI server adapter with slot number 60 is added for the virtual tape and one client SCSI adapter with slot number 60 is added to the AIX V6.1 and AIX V5.3 partitions (not shown for the sake of simplicity).

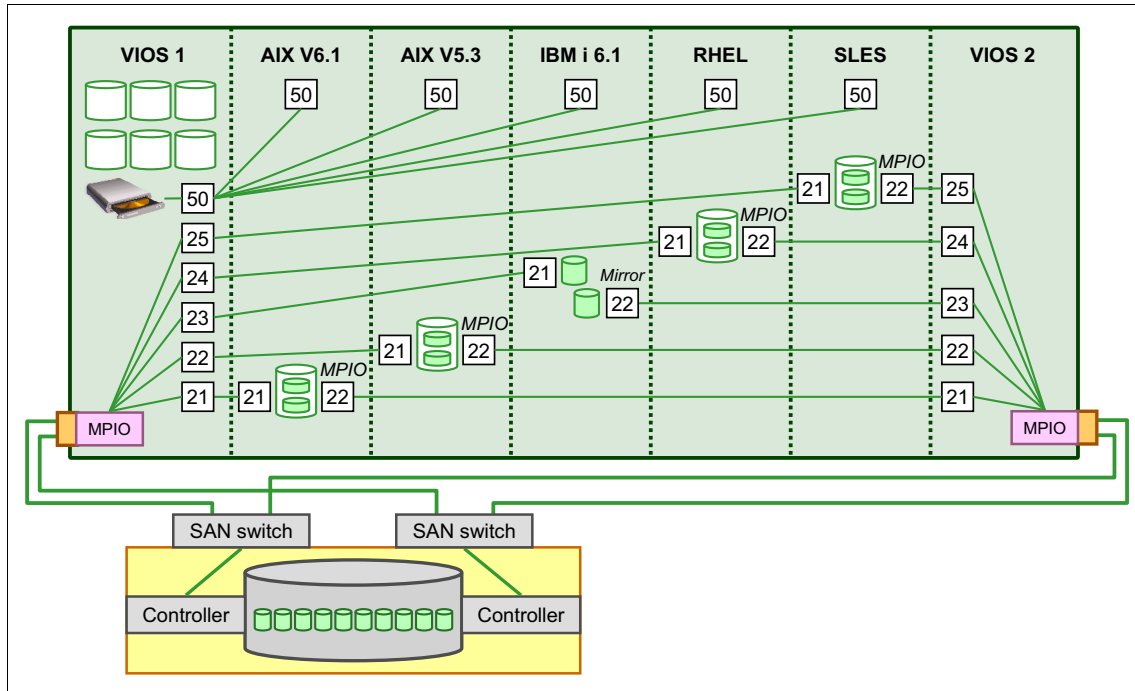


Figure H-1 The partition and slot numbering plan of virtual storage adapters

Tip: You are not required to have the same slot numbering for the server and client adapters: it simply makes it easier to keep track.

Figure H-2 shows the basic layout of partitions and the slot numbering of virtual Ethernet adapters.

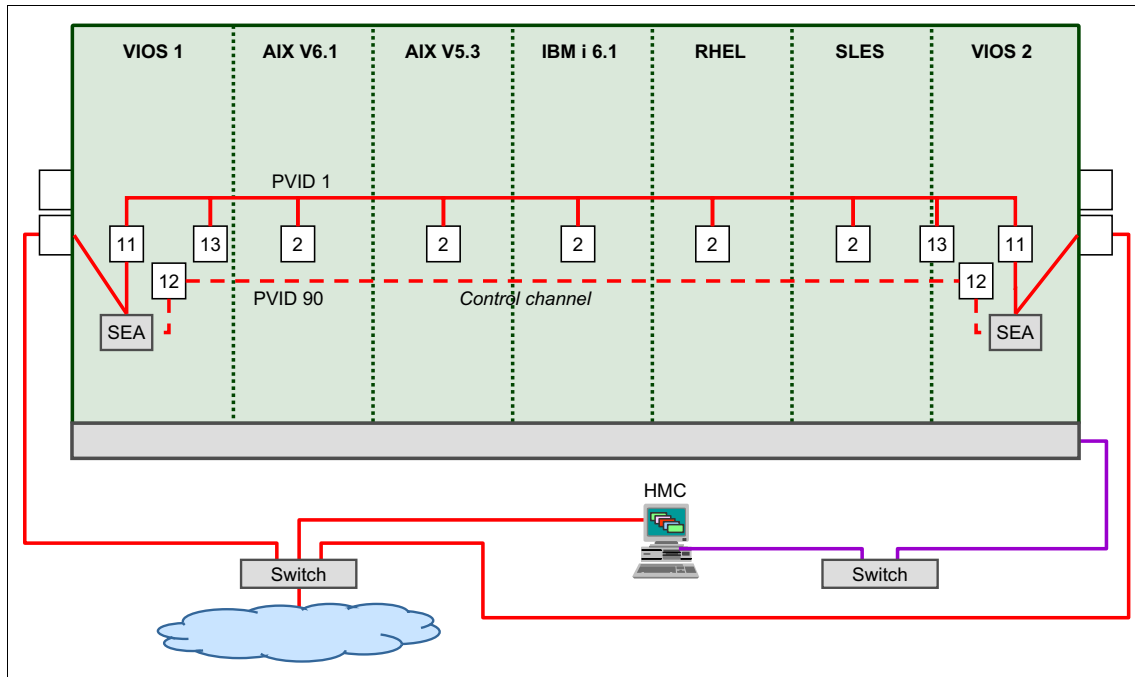


Figure H-2 The partition and slot numbering plan for virtual Ethernet adapters

H.2 Preparation recommendations

When you deploy a System Plan on the HMC, the configuration is validated against the installed system (adapter and disk features and their slot numbers, amount of physical and CoD memory, number and type of physical and CoD CPUs, and more). To simplify matching the SPT System Plan and the physical system, follow these steps:

1. Create a System Plan of the physical system on the HMC or IVM.
2. Export the System Plan to SPT on your PC and convert it to SPT format.

Tip: If the System Plan cannot be converted, use the System Plan to manually create the compatible configuration.

3. Use this System Plan as a template and customize it to meet your requirements.
4. Import the completed SPT System Plan to the HMC or IVM and deploy it.

H.3 Planning the configuration with SPT

Figure H-3 shows the *Partition properties* window where you can add or modify partition profiles. Notice the *Processor* and *Memory* tabs for setting those properties.

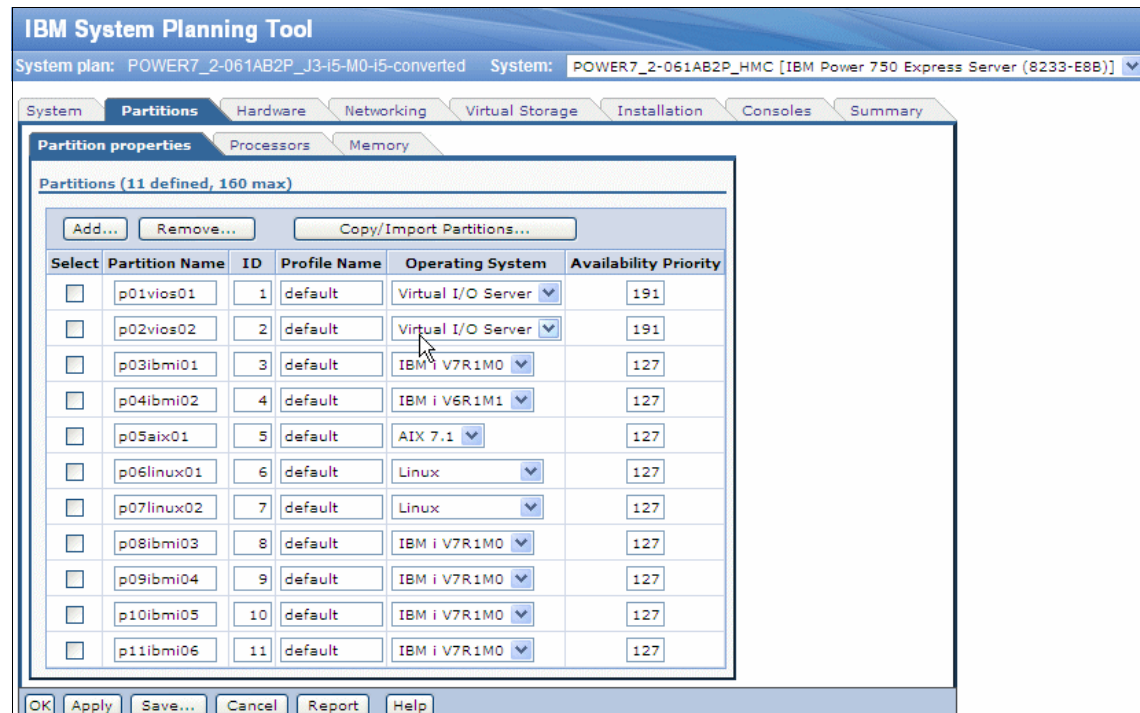


Figure H-3 The SPT Partition properties window

A useful feature in SPT is the ability to edit virtual adapter slot numbers and check consistency for server-client slot numbers. As a general rule, increase the maximum number of virtual slots for the partitions.

Figure H-4 shows the *Virtual SCSI* window. Click **Edit Virtual Slots** to open the window.

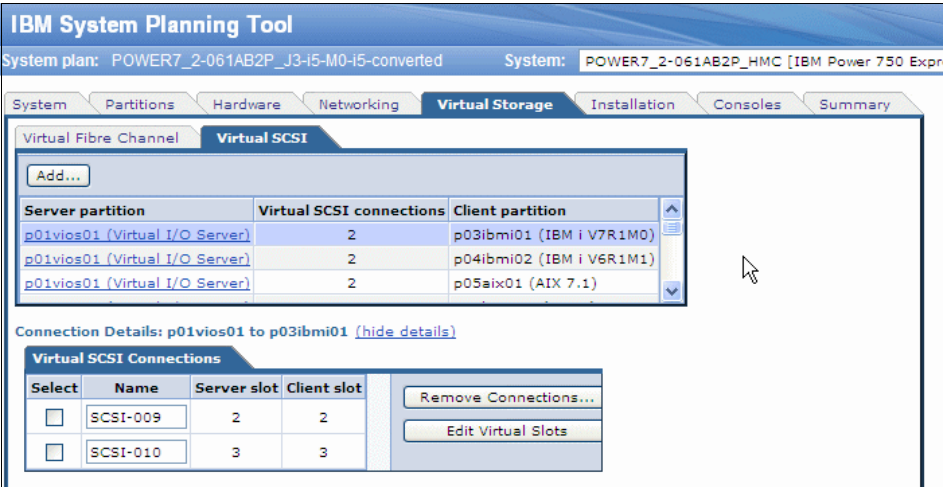


Figure H-4 The SPT Virtual SCSI window

5. Figure H-5 shows the **Edit Virtual Slots** window. Here the two Virtual I/O Servers are listed on the left side and the client partitions are listed on the right side. The maximum number of virtual adapters is 1024 for the Virtual I/O Servers. In the SCSI area, you can check that the server adapters from the Virtual I/O Servers match the client partitions.

System: POWER7_2-061AB2P_HMC [IBM Power 750 Express Server (8233-E8B)]

Edit Virtual Slots

Work with virtual adapter slots and mappings

p01vios01 (Virtual I/O Server)

Total slots: **1024** Used slots: 21 Available slots: 1003

Console

Slot	Type	Target Partition	Target Slot
0	Server		
1	Server		

Ethernet

Slot	PVID	Additional VLANs
10	1	

SCSI

Slot	Type	Target Partition	Target Slot
103	Server	p03ibmi01 (IBM i V7R1M0)	11
104	Server	p04ibmi02 (IBM i V6R1M1)	11
105	Server	p05aix01 (AIX 7.1)	11
106	Server	p06linux01 (Linux)	11
107	Server	p07linux02 (Linux)	11
108	Server	p08ibmi03 (IBM i V7R1M0)	11
109	Server	p09ibmi04 (IBM i V7R1M0)	11
110	Server	p10ibmi05 (IBM i V7R1M0)	11

p08ibmi03 (IBM i V7R1M0)

Total slots: **40** Used slots: 7 Available slots: 33

Console

Slot	Type	Target Partition	Target Slot
0	Server		
1	Server		

Ethernet

Slot	PVID	Additional VLANs
10	1	

SCSI

Slot	Type	Target Partition	Target Slot
11	Client	p01vios01 (Virtual I/O Server)	108
19	Client	p01vios01 (Virtual I/O Server)	908
21	Client	p02vios02 (Virtual I/O Server)	108
29	Client	p02vios02 (Virtual I/O Server)	908

Figure H-5 The SPT Edit Virtual Slots window

Note: If your configuration includes the use of Virtual Fibre Channel, their slot numbers can also be edited on this window.

- When all elements of the System Plan are completed, you can import it to the HMC to be deployed. Figure H-6 shows the HMC with the imported SPT System Plan ready to be deployed.

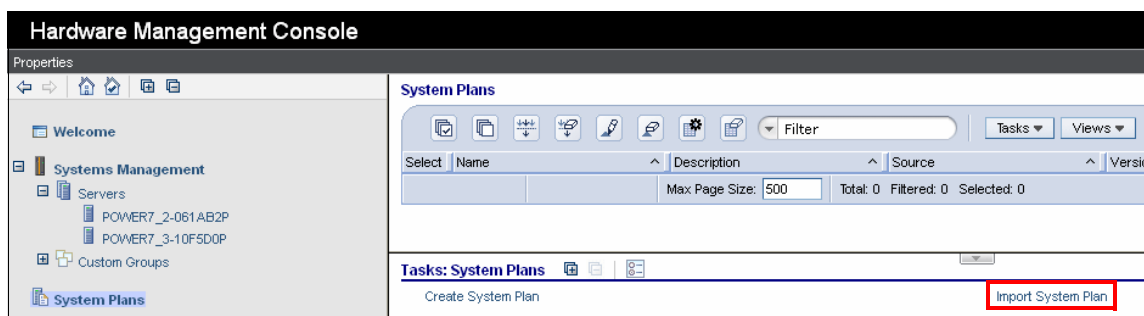


Figure H-6 System Planning Tool ready to be deployed

- Select **Deploy System Plan** as shown in Figure H-7 to start the Deploy System Plan Wizard.



Figure H-7 Deploy System Plan

- Figure H-8 shows the first menu window where you can select the System Plan and target managed system.

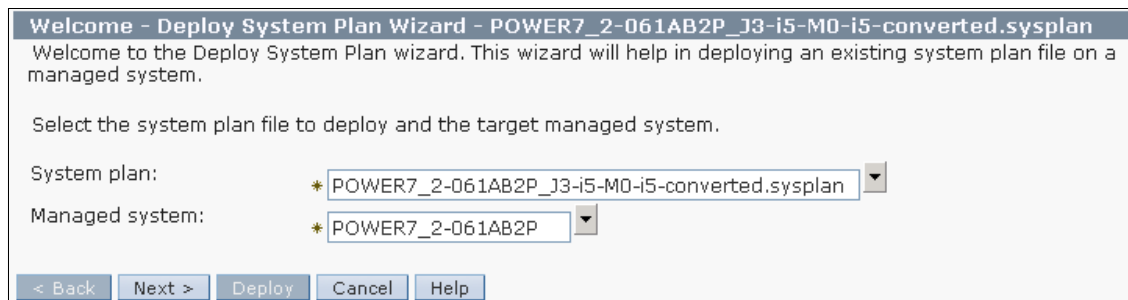


Figure H-8 Deploy System Plan Wizard

9. The next step is validation. The selected System Plan is validated against the target managed system. The validation window is shown in Figure H-9.

Validation - Deploy System Plan Wizard - POWER7_2-061AB2P_J3-i5-M0-i5-converted.sysplan
Hardware and Partition Validation are complete for system plan 'POWER7_2-061AB2P_J3-i5-M0-i5-converted.sysplan' and managed system 'POWER7_2-061AB2P'

Validation Progress

Validation Type	Status
Hardware validation	Successful
Partition validation	Successful

Figure H-9 The System Plan validation window

10. Next, the partition profiles to be deployed are selected as shown in Figure H-10. In this case, all partition profiles will be deployed.

Partition Deployment - Deploy System Plan Wizard - POWER7_2-061AB2P_J3-i5-M0-i5-converted.sysplan
Use this page to specify which partition plan actions to deploy on the managed system. Only the checked plan actions will be deployed. Select a row in the Partition Plan Actions table to view more details about the partition plan action.

Partition Plan Actions

Select	Dependency Hierarchy	Plan Action	Deploy	Status
☐	1.1	Partition p01vios01	☒	
☐	1.2	Partition p02vios02	☒	
☐	1.3	Partition p03ibmi01	☒	
☐	1.4	Partition p04ibmi02	☒	
☐	1.5	Partition p05aix01	☒	
☐	1.6	Partition p06linux01	☒	
☐	1.7	Partition p07linux02	☒	
☐	1.8	Partition p08ibmi03	☒	
☐	1.9	Partition p09ibmi04	☒	
☐	1.10	Partition p10ibmi05	☒	

Details

Partition Deployment Step Order
This table displays the partition deployment steps that will be performed based on the items checked in the Partition Plan Actions table.

Deployment Step
Partition p01vios01
Partition p02vios02
Partition p03ibmi01
Partition p04ibmi02
Partition p05aix01
Partition p06linux01
Partition p07linux02
Partition p08ibmi03
Partition p09ibmi04
Partition p10ibmi05

< Back Next > Deploy Cancel Help

Figure H-10 Partitions to Deploy

11. Next window shows the Deployment Step Order as in Figure H-11. Click **Next** to continue to the deployment progress steps.

Summary - Deploy System Plan Wizard - POWER7_2-061AB2P_J3-i5-M0-i5-converted.sysplan
You are now ready to deploy the system plan. Click Deploy to deploy the system plan in the order shown below.

System plan: POWER7_2-061AB2P_J3-i5-M0-i5-converted.sysplan
Managed system: POWER7_2-061AB2P

Deployment Step Order

Deployment Step
Backup existing partition configuration data
Partition p01vios01
Partition p02vios02
Partition p03ibmi01
Partition p04ibmi02
Partition p05aix01
Partition p06linux01
Partition p07linux02
Partition p08ibmi03
Partition p09ibmi04

Note
Please DO NOT perform any other actions on this managed system while deployment is running.

< Back Next > Deploy Cancel Help

Figure H-11 The Deployment Steps

Important: The HMC must be prepared with the correct resources for operating system installation.

You can load the Virtual I/O Server operating system onto the HMC from DVD or NIM using the **0S_insta11** command.

You can define it as a resource using the **defsysplanres** command, or using the HMC graphical user interface by clicking **HMC Management** → **Manage Install resources**.

12. When the preparation steps have been completed, click **Deploy** to start the deployment. The deployment progress is logged as shown in Figure H-12.

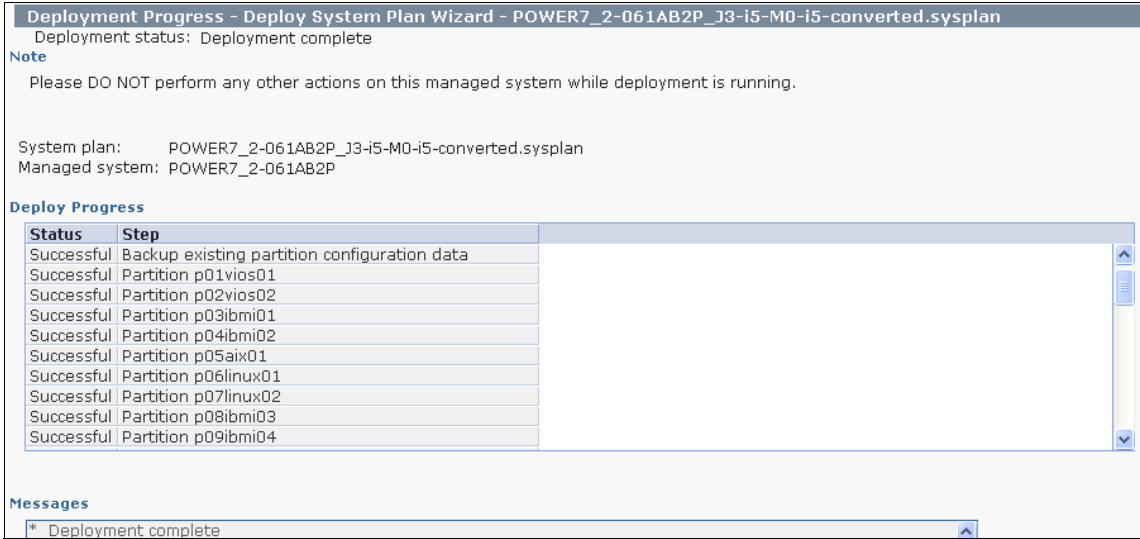


Figure H-12 The Deployment Progress window

Figure H-13 shows the partition profiles when deployed on the HMC. These partitions are now ready for installation of the operating systems.

Select	Name	ID	Status	Processing Units	Memory (GB)	Active Profile	Environment	Reference Code
☐	p01vios01	1	Not Activated	0	0	0	Virtual I/O Server	00000000
☐	p02vios02	2	Not Activated	0	0	0	Virtual I/O Server	00000000
☐	p03lmi01	3	Not Activated	0	0	0	IBM i	00000000
☐	p04lmi02	4	Not Activated	0	0	0	IBM i	00000000
☐	p05aix01	5	Not Activated	0	0	0	AIX or Linux	00000000
☐	p06linux01	6	Not Activated	0	0	0	AIX or Linux	00000000
☐	p07linux02	7	Not Activated	0	0	0	AIX or Linux	00000000
☐	p08lmi03	8	Not Activated	0	0	0	IBM i	00000000
☐	p09lmi04	9	Not Activated	0	0	0	IBM i	00000000
☐	p10lmi05	10	Not Activated	0	0	0	IBM i	00000000
☐	p11lmi06	11	Not Activated	0	0	0	IBM i	00000000

Figure H-13 Partition profiles deployed on the HMC

All profiles are created with physical and virtual adapter assignments. In this scenario, the operating system for the Virtual I/O Servers or any of the client partitions was not installed in the deployment process. After the System Plan has been deployed and the configuration has been customized, a System Plan should be created from the HMC.

The HMC generated System Plan provides excellent documentation of the installed system. This System Plan can also be used as a backup of the managed system configuration.

Important: The first page of the System Plan might be marked as follows:



This system plan is not valid for deployment.

This means that it *cannot* be used to restore the configuration.

In case a System Plan cannot be generated using the HMC graphical user interface, you can use the following HMC command:

```
HMC restricted shell> mksysplan -m <managed system> -f\ <filename>.sysplan --noprobe
```

H.4 Initial setup checklist

This section contains a high level listing of common steps for the initial setup of a new system using SPT. Customize the list to fit your environment:

1. Make a System Plan from the HMC of the new system.
Delete the pre-installed partition if the new system comes with such a partition.
This System Plan is a baseline for configuring the new system. It will have the adapter's slot assignment, CPU, and memory configurations.
2. Export the System Plan from the HMC into SPT.
In SPT, the file must be converted to SPT format.
3. Complete the configuration as much as possible in SPT:
 - ☐ Add one Virtual I/O Server partition if using virtual I/O.
 - ☐ Add one more Virtual I/O Server for a dual configuration, if required. Dual Virtual I/O Server provides higher serviceability.
 - ☐ Add the client partition profiles.
 - ☐ Assign CPU and memory resources to all partitions.
 - ☐ Create the required configurations for storage and network in SPT.
 - ☐ Add virtual storage as local disks or SAN disks.
 - ☐ Configure SCSI connections for MPIO or mirroring if you are using a dual Virtual I/O Server configuration.
 - ☐ Configure virtual networks and SEA for attachment to external networks.
 - ☐ For a dual Virtual I/O Server configuration, configure SEA failover, or Network Interface Backup (NIB) as appropriate for virtual network redundancy.
 - ☐ Create a virtual server adapter for virtual DVD and for virtual tape if a tape drive is installed.
 - ☐ Apply your slot numbering structure according to your plan.
4. Import the SPT System Plan into the HMC and deploy it to have the profiles generated. Alternatively, profiles can be generated directly on the HMC.
5. If using SAN disks, create and map them to the host or host group of the Fibre Channel adapters.
 - ☐ If using Dual Virtual I/O Servers, the `reserve_policy` must be changed from `single_path` to `no_reserve`.

- ☐ SAN disks must be mapped to all Fibre Channel adapters that will be target in Partition Mobility.
- 6. Install the first Virtual I/O Server from DVD or NIM.
 - ☐ Upgrade the Virtual I/O Server if updates are available.
 - ☐ Mirror the rootvg disk.
 - ☐ Create or install SSH keys. The SSH subsystem is installed in the Virtual I/O Server by default.
 - ☐ Configure time protocol services.
 - ☐ Add users.
 - ☐ Set the security level and firewall settings if required.
- 7. Configure an internal network connected to the external network by configuring a Shared Ethernet Adapter (SEA).
 - ☐ Consider adding a separate virtual adapter to the Virtual I/O Server to carry the IP address instead of assigning it to the SEA.
- 8. Create a backup of the Virtual I/O Server to local disk by using the **backupios** command.
- 9. Map disks to the client partitions with the **mkvdev** command.
 - ☐ Map local disks or local partitions.
 - ☐ Map SAN disks through virtual SCSI adapter.
 - ☐ Map SAN disks through virtual Fibre Channel adapter.
- 10. Map the DVD drive to a virtual DVD for the client partitions by using the **mkvdev** command.
- 11. If available, map the tape drive to a virtual tape drive for the client partitions by using the **mkvdev** command.
- 12. Add a client partition to be a NIM server to install the AIX and Linux partitions. If a NIM server is already available, skip this step.
 - ☐ Boot a client partition to SMS and install AIX from the virtual DVD.
 - ☐ Configure NIM on the client partition.
 - ☐ Let the NIM resources reside in a separate volume group. The rootvg volume group should be kept as compact as possible.
- 13. Copy the base mksysb image to the NIM server and create the required NIM resources.
- 14. If using dual Virtual I/O Servers, perform a NIM install of the second Virtual I/O Server from the base backup of the first Virtual I/O Server. If a single Virtual I/O Server is used, go directly to step number 20.

15. Configure the second Virtual I/O Server.
16. Map disks from the second Virtual I/O Server to the client partitions.
17. Configure SEA failover for network redundancy on the first Virtual I/O Server.
18. Configure SEA failover for network redundancy on the second Virtual I/O Server.
19. Test that SEA failover is operating correctly.
20. Install the operating system on the client partitions using NIM or the virtual DVD.
 - ☐ Configure NIB if this is used for network redundancy.
 - ☐ If using MPIO, change the hcheck_interval parameter with the **chdev** command to have the state of paths updated automatically.
 - ☐ Test that NIB failover is configured correctly in client partitions if NIB is used for network redundancy.
 - ☐ Test mirroring if this is used for disk redundancy.
21. Create a system backup of both Virtual I/O Servers using the **backupios** command.
22. Document the Virtual I/O Server environment.
 - ☐ List virtual SCSI, Fibre Channel, and network mappings with the **lsmap** command.
 - ☐ List network definitions.
 - ☐ List security settings.
 - ☐ List user definitions.
23. Create a system backup of all client partitions.
24. Create a System Plan of the installed configuration from the HMC as documentation and backup.
25. Save the profiles on the HMC. Select the Managed System and click **Configuration** → **Manage Partition Data** → **Backup**. Enter the name of your profile backup. This backup is a record of all partition profiles on the Managed System.
26. Back up HMC information to DVD, to a remote system, or to a remote site. Click **HMC Management** → **Back up HMC Data**. In the menu, select a target for the backup and follow the provided instructions. The backup contains all HMC settings such as user, network, security and profile data.
27. Start collecting performance data. It is valuable to collect long-term performance data to have a baseline of performance history.

Abbreviations and acronyms

ABI	Application Binary Interface	CHRP	Common Hardware Reference Platform
AC	Alternating Current	CLI	Command Line Interface
ACL	Access Control List	CLVM	Concurrent LVM
AFPA	Adaptive Fast Path Architecture	CPU	Central Processing Unit
AIO	Asynchronous I/O	CRC	Cyclic Redundancy Check
AIX	Advanced Interactive Executive	CSM	Cluster Systems Management
APAR	Authorized Program Analysis Report	CUoD	Capacity Upgrade on Demand
API	Application Programming Interface	DCM	Dual Chip Module
ARP	Address Resolution Protocol	DES	Data Encryption Standard
ASMI	Advanced System Management Interface	DGD	Dead Gateway Detection
BFF	Backup File Format	DHCP	Dynamic Host Configuration Protocol
BIND	Berkeley Internet Name Domain	DLPAR	Dynamic LPAR
BIST	Built-In Self-Test	DMA	Direct Memory Access
BLV	Boot Logical Volume	DNS	Domain Naming System
BOOTP	Boot Protocol	DRM	Dynamic Reconfiguration Manager
BOS	Base Operating System	DR	Dynamic Reconfiguration
BSD	Berkeley Software Distribution	DVD	Digital Versatile Disk
CA	Certificate Authority	EC	EtherChannel
CATE	Certified Advanced Technical Expert	ECC	Error Checking and Correcting
CD	Compact Disk	EOF	End of File
CDE	Common Desktop Environment	EPOW	Environmental and Power Warning
CD-R	CD Recordable	ERRM	Event Response resource manager
CD-ROM	Compact Disk-Read Only Memory	ESS	Enterprise Storage Server
CEC	Central Electronics Complex	F/C	Feature Code
		FC	Fibre Channel
		FCAL	Fibre Channel Arbitrated Loop

FDX	Full Duplex	LA	Link Aggregation
FLOP	Floating Point Operation	LACP	Link Aggregation Control Protocol
FRU	Field Replaceable Unit	LAN	Local Area Network
FTP	File Transfer Protocol	LDAP	Lightweight Directory Access Protocol
GDPS®	Geographically Dispersed Parallel Sysplex™	LED	Light Emitting Diode
GID	Group ID	LMB	Logical Memory Block
GPFS™	General Parallel File System	LPAR	Logical Partition
GUI	Graphical User Interface	LPP	Licensed Program Product
HACMP™	High Availability Cluster Multiprocessing	LUN	Logical Unit Number
HBA	Host Bus Adapters	LV	Logical Volume
HMC	Hardware Management Console	LVCB	Logical Volume Control Block
HTML	Hypertext Markup Language	LVM	Logical Volume Manager
HTTP	Hypertext Transfer Protocol	MAC	Media Access Control
Hz	Hertz	Mbps	Megabits Per Second
I/O	Input/Output	MBps	Megabytes Per Second
IBM	International Business Machines	MCM	Multichip Module
ID	Identification	ML	Maintenance Level
IDE	Integrated Device Electronics	MP	Multiprocessor
IEEE	Institute of Electrical and Electronics Engineers	MPIO	Multipath I/O
IP	Internetwork Protocol	MTU	Maximum Transmission Unit
IPAT	IP Address Takeover	NFS	Network File System
IPL	Initial Program Load	NIB	Network Interface Backup
IPMP	IP Multipathing	NIM	Network Installation Management
ISV	Independent Software Vendor	NIMOL	NIM on Linux
ITSO	International Technical Support Organization	NVRAM	Non-Volatile Random Access Memory
IVM	Integrated Virtualization Manager	ODM	Object Data Manager
JFS	Journaled File System	OSPF	Open Shortest Path First
L1	Level 1	PCI	Peripheral Component Interconnect
L2	Level 2	PIC	Pool Idle Count
L3	Level 3	PID	Process ID
		PKI	Public Key Infrastructure
		PLM	Partition Load Manager

POST	Power-On Self-test	SAN	Storage Area Network
POWER	Performance Optimization with Enhanced Risc (Architecture)	SCSI	Small Computer System Interface
PPC	Physical Processor Consumption	SDD	Subsystem Device Driver
PPFC	Physical Processor Fraction Consumed	SMIT	System Management Interface Tool
PTF	Program Temporary Fix	SMP	Symmetric Multiprocessor
PTX	Performance Toolbox	SMS	System Management Services
PURR	Processor Utilization Resource Register	SMT	Simultaneous Multithreading
PV	Physical Volume	SP	Service Processor
PVID	Physical Volume Identifier	SPOT	Shared Product Object Tree
PVID	Port Virtual LAN Identifier	SRC	System Resource Controller
QoS	Quality of Service	SRN	Service Request Number
RAID	Redundant Array of Independent Disks	SSA	Serial Storage Architecture
RAM	Random Access Memory	SSH	Secure Shell
RAS	Reliability, Availability, and Serviceability	SSL	Secure Socket Layer
RBAC	Role bases access control	SUID	Set User ID
RCP	Remote Copy	SVC	SAN Virtualization Controller
RDAC	Redundant Disk Array Controller	TCP/IP	Transmission Control Protocol/Internet Protocol
RIO	Remote I/O	TSA	Tivoli System Automation
RIP	Routing Information Protocol	TL	Technology Level
RISC	Reduced Instruction-Set Computer	UDF	Universal Disk Format
RMC	Resource Monitoring and Control	UDID	Universal Disk Identification
RPC	Remote Procedure Call	VIPA	Virtual IP Address
RPL	Remote Program Loader	VG	Volume Group
RPM	Red Hat Package Manager	VGDA	Volume Group Descriptor Area
RSA	Rivet, Shamir, Adelman	VGSA	Volume Group Status Area
RSCT	Reliable Scalable Cluster Technology	VLAN	Virtual Local Area Network
RSH	Remote Shell	VP	Virtual Processor
		VPD	Vital Product Data
		VPN	Virtual Private Network
		RRRP	Virtual Router Redundancy Protocol
		VSD	Virtual Shared Disk

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks publications

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only.

- ▶ *IBM PowerVM Virtualization Managing and Monitoring*, SG24-7590
- ▶ *IBM Systems Director VMControl Implementation Guide on IBM Power Systems*, SG24-7829
- ▶ *Implementing an IBM b-type SAN with 8 Gbps Directors and Switches*, SG24-6116
- ▶ *Integrated Virtual Ethernet Adapter Technical Overview and Introduction*, REDP-4340
- ▶ *Power Systems Enterprise Servers with PowerVM Virtualization and RAS*, SG24-7965
- ▶ *Power Systems Memory Deduplication*, REDP-4827
- ▶ *Virtual Partition Manager A Guide to Planning and Implementation*, REDP-4013
- ▶ *IBM Power Systems HMC Implementation and Usage Guide*, SG24-7491

You can search for, view, download or order these documents and other Redbooks publications, Redpaper publications, Web Docs, draft and additional materials, at the following website:

ibm.com/redbooks

Online resources

These websites are also relevant as further information sources:

- ▶ Availability of the PowerVM features by Power Systems models
<http://www.ibm.com/systems/power/software/virtualization/editions/features.html>
- ▶ Power Systems Technical Guide
<http://www.ibm.com/systems/power/hardware/reports/factsfeatures.html>
- ▶ Capacity on Demand activation code
<http://www-912.ibm.com/pod/pod>
- ▶ IBM POWER7 PowerVM Active Memory Sharing
http://pic.dhe.ibm.com/infocenter/powersys/v3r1m5/topic/p7eew/p7eew_ams.htm
- ▶ IBM Virtual I/O Server datasheet
<http://www14.software.ibm.com/webapp/set2/sas/f/vios/documentation/datasheet.html>
- ▶ IBM System Storage Interoperation Center (SSIC)
<http://www-03.ibm.com/systems/support/storage/config/ssic>
- ▶ Tivoli Storage Manager Overview
<http://www-306.ibm.com/software/tivoli/products/storage-mgr/>
- ▶ Tivoli Application Dependency Discovery Manager
<http://www-306.ibm.com/software/tivoli/products/taddm/>
- ▶ IBM SmartCloud Cost Management
<http://www-306.ibm.com/software/tivoli/products/usage-accounting/>
- ▶ IBM Security Identity Manager
<http://www-306.ibm.com/software/tivoli/products/identity-mgr/>
- ▶ Tivoli Monitoring
<http://www-01.ibm.com/software/tivoli/products/monitor>
- ▶ Tivoli Security Compliance Manager
<http://www-01.ibm.com/software/tivoli/products/security-compliance-mgr>

- ▶ IBM Virtual I/O Server allowed third-party applications
http://www.ibm.com/partnerworld/gsd/searchprofile.do?name=VIOS_Recognized_List
- ▶ IBM i Virtualization and Open Storage Read-me first
http://www-03.ibm.com/systems/resources/systems_i_Virtualization_Open_Storage.pdf
- ▶ IBM Service and productivity tools for POWER Linux servers
<http://www14.software.ibm.com/webapp/set2/sas/f/lopdiags/home.html>
- ▶ IBM Systems Director 6.3.1 Information Center
<http://publib.boulder.ibm.com/infocenter/director/pubs/>
- ▶ FTP site to download latest Linux kernel source
<ftp://ftp.kernel.org/pub/linux/kernel/>
- ▶ Availability of PowerVM features by Power Systems models
<http://www.ibm.com/systems/power/software/virtualization/editions/features.html>
- ▶ IBM i Live Partition Mobility download page
http://www-912.ibm.com/s_dir/SLKBase.nsf/1ac66549a21402188625680b0002037e/e1877ed7f3b0cfa8862579ec0048e067?OpenDocument#_Section1
- ▶ IBM PowerHA SystemMirror 7.1
<http://pic.dhe.ibm.com/infocenter/aix/v7r1/index.jsp?topic=%2Fcom.ibm.aix.powerha.navigation%2Fpowerha.htm>
- ▶ IBM PowerSC Overview
<http://www-03.ibm.com/systems/power/software/security/>
- ▶ IBM PowerSC AIX 7.1 information
http://pic.dhe.ibm.com/infocenter/aix/v7r1/topic/com.ibm.aix.doc/doc/base/powersc_main.htm
- ▶ IBM PowerSC Platform offerings
<http://www.ibm.com/systems/power/software/security/offerings.html>
- ▶ IBM Hardware Management Console information
http://pic.dhe.ibm.com/infocenter/powersys/v3r1m5/topic/ipha8/hwparent_hmc.htm
- ▶ IBM Fix Central
<http://www.ibm.com/support/fixcentral/>

- ▶ IBM Systems Workload Estimator
<http://www.ibm.com/systems/support/tools/estimator/index.html>
- ▶ IBM POWER7 Capacity planning information
http://publib.boulder.ibm.com/infocenter/powersys/v3r1m5/topic/p7hb1/iphb1_vios_planning_cap.htm
- ▶ IBM POWER7 Virtual I/O Server information
<http://publib.boulder.ibm.com/infocenter/powersys/v3r1m5/index.jsp?topic=/p7hb1/iphb1kickoff.htm>
- ▶ IBM POWER7 Device compatibility in a Virtual I/O Server environment
http://publib.boulder.ibm.com/infocenter/powersys/v3r1m5/topic/p7hb1/iphb1_vios_device_compat.htm
- ▶ IBM BladeCenter Interoperability Guide download page
<http://www-947.ibm.com/support/entry/portal/docdisplay?brand=5000020&indocid=MIGR-5073016>
- ▶ IBM i NPIV System and Software Requirements download page
<http://www-01.ibm.com/support/docview.wss?uid=nas13b3ed3c69d4b7f25862576b700710198>
- ▶ IBM Virtual I/O Server support information
<http://www14.software.ibm.com/support/customercare/sas/f/vios/home.html>
- ▶ IBM Virtual I/O Server support datasheet
<http://www14.software.ibm.com/webapp/set2/sas/f/vios/documentation/datasheet.html>
- ▶ IBM POWER6 Device compatibility in a Virtual I/O Server environment
http://publib.boulder.ibm.com/infocenter/powersys/v3r1m5/index.jsp?topic=/iphb1/iphb1_vios_device_compat.htm
- ▶ Installing IBM i and related software on a new system or logical partition
<http://publib.boulder.ibm.com/infocenter/series/v7r1m0/index.jsp?topic=/rzahc/rzahcinstall.htm>
- ▶ IBM Installation Toolkit for PowerLinux
<http://www14.software.ibm.com/webapp/set2/sas/f/lopdiags/installtools>
- ▶ Novell SUSE Linux Multipath Management Tools
http://www.novell.com/documentation/sles10/stor_admin/?page=/documentation/sles10/stor_admin/data/mpiotools.html

- ▶ IBM POWER Linux Service and productivity tools
<https://www14.software.ibm.com/webapp/set2/sas/f/lopdiags/home.html>
- ▶ IBM SCSI - Hot add, remove, rescan of SCSI devices
<http://www-941.ibm.com/collaboration/wiki/display/LinuxP/SCSI+-+Hot+add%2C+remove%2C+rescan+of+SCSI+devices>
- ▶ IBM Fix Level Recommendation Tool
<https://www14.software.ibm.com/webapp/set2/flrt/home>
- ▶ Configuring Resource Monitoring and Control (RMC) for the Partition Load Manager
http://publib.boulder.ibm.com/infocenter/systems/scope/hw/index.jsp?topic=iphbk/iphbkrmc_configuration.htm
- ▶ IBM System Director VMControl
<http://www.ibm.com/systems/software/director/vmcontrol/>
- ▶ IBM PowerLinux Overview
<http://www.ibm.com/systems/power/software/linux>
- ▶ IBM Power ISA Version 2.05
https://power.org/wp-content/uploads/2012/07/PowerISA_V2.05.pdf
- ▶ IBM Power ISA Version 2.06 Revision B
https://power.org/wp-content/uploads/2012/07/PowerISA_V2.06B_V2_PUBLIC.pdf
- ▶ Power System Capacity on Demand solutions for System i and System p
<http://www.ibm.com/systems/power/hardware/cod>
- ▶ IBM System Planning Tool
<http://www.ibm.com/systems/support/tools/systemplanningtool>
- ▶ IBM EnergyScale for POWER7 Processor-Based Systems
<http://www.ibm.com/systems/power/hardware/whitepapers/energyscale7.html>
- ▶ IBM Passport Advantage Licensing
<http://www-306.ibm.com/software/lotus/passportadvantage/licensing.html>
- ▶ IBM Passport Advantage Licensing program overview
<http://www.ibm.com/software/passportadvantage>

- ▶ Novell SUSE Linux Enterprise Server
<http://www.novell.com/products/server/>
- ▶ IBM Red Hat
<https://www.redhat.com/>
- ▶ IBM i Access Overview
<http://www-03.ibm.com/systems/i/software/access/caann.html>
- ▶ Tivoli Directory Server
<http://www-01.ibm.com/software/tivoli/products/directory-server>
- ▶ IBM AIX Installing the Global Security Kit
http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.itame.doc_5.1/am51_webinstall223.htm
- ▶ IBM i 7.1 InformationCenter
<http://publib.boulder.ibm.com/infocenter/iseriess/v7r1m0/index.jsp>
- ▶ IBM Storage are network (SAN)
<http://www-03.ibm.com/systems/storage/san/index.html>
- ▶ IBM Bonding configuration
<http://www.ibm.com/developerworks/wiki/display/LinuxP/Bonding+configuration>
- ▶ IBM POWER7 Installing the Virtual I/O Server from the HMC
http://publib.boulder.ibm.com/infocenter/powersys/v3r1m5/index.jsp?topic=/p7hb1/iphb1_vios_configuring_installhmc.htm
- ▶ IBM AIX 6.1 Information Center
<http://publib.boulder.ibm.com/infocenter/pseries/v6r1/index.jsp>
- ▶ IBM Power Systems Hardware Information Center
<http://publib.boulder.ibm.com/infocenter/powersys/v3r1m5/index.jsp>

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

A

- access external network flag 238
- accounting, AIX 294
- Active Memory Deduplication 27
 - planning 157
 - setup 414
- Active Memory Expansion 29–30
 - compressed pool 697
 - DLPAR 702
 - enabling 700
 - feature code 695
 - memory deficit 698
 - memory expansion factor 696
 - requirements 696
 - setup 699
 - uncompressed pool 697
- Active Memory Sharing 30
 - setup 402
- active migration 264–265
 - capability 262
 - compatibility 262
 - entitlements 263
 - example 267
 - migratability 263
 - multiple concurrent migrations 286
 - prerequisites 263
 - processor compatibility modes 292
 - remote 288
 - requirements 287
 - Shared Ethernet Adapter 266
 - time 287
- adapter
 - dedicated 263
 - physical 263
- advanced accounting 294
- Advanced System Management Interface, ASMI 15, 619, 624
- AIX 294
 - boot image 688
 - EtherChannel Backup, ECB 603
 - mirroring 540
 - multipathing 511
 - SMIT 690

- alert command 59
- amepat command 698–699
- application WPARs 710
- ARP 229
- assigned processor 134
- Authorizations corresponding to Virtual I/O Server commands 455
- autoyast 379
- availability 103

B

- backupos command 348, 725
- barrier synchronization register, BSR 264, 620, 638
- battery power 263, 624
- bindprocessor command 116
- boot image 688
- boot mode, SMS 339
- bootlist command 541
- bosboot command 541, 688
- broadcast 229

C

- cache 686
- capacity
 - upper boundary 115
- Capacity on Demand 626, 629, 683
 - licensing 137
- capped micro-partition 113, 116–117, 256
- cfgassist command 420, 429, 590, 595
- cfgdev command 345, 491–492
- cfgmgr command 49, 491–492
- changing SEA failover in load sharing mode 600
- characteristics of WPARs 709
- chauth command 463
- chdev command 494, 496, 499, 503, 579, 645–646, 726
- chhwres command 52, 192, 395
- chkdev command 183
- chlparstate command 660, 667
- chpath command 494, 514
- chrole command 463
- chsp command 574
- chsyscfg command 656

- chuser command 450
- cleargcl command 451
- client adapter
 - virtual SCSI 48
- cluster
 - create 571
 - repository 571–572
- cluster command 571
- commands
 - AIX 579
 - amepat 698–699
 - arp 229
 - bindprocessor 116
 - bootlist 541
 - bosboot 541, 688
 - cfgmgr 49, 491–492
 - chdev 494, 496, 726
 - chpath 494, 514
 - errpt 542
 - ethchan_config 608
 - expansion 445
 - filemon 294
 - ldapsearch 444, 446
 - lparstat 701
 - lspp command 445
 - lspath 511, 515
 - lspv 511, 542
 - mirrorvg 540
 - rset 116
 - smctl 688
 - smitty 690
 - smitty installios 334
 - smtctl 688
 - syncvg 542
 - tcptr 616
 - topas 294
 - topasrec 679
 - tprof 294
 - varyonvg 204, 542
 - HMC
 - chhwres 52, 192, 395
 - chlparstate 660, 667
 - chsyscfg 656
 - defsysplanres 721
 - installios 342
 - lshwres 396, 654
 - lsiparmigr 273, 286
 - lssyscfg 657
 - migrpar 287
 - mksyscfg 52, 192
 - mksysplan 723
 - OS_install 721
 - IBM i
 - QPRCMLTTSK 691
 - Linux
 - arp 229
 - brctl 229
 - dmesg 489
 - fdisk 498
 - lsmod 489
 - lsscsi 489, 498, 533
 - mdadm 204, 208, 570
 - mt 493
 - multipath 499, 531
 - ppc64_cpu 692
 - vconfig 75
 - NIM
 - lsnim 368
 - other
 - gsk7ikm 432
 - mkldap 445
 - mksecdap 439
 - SAN switch
 - portCfgNPIVPort 476
 - portcfgshow 476
 - version 475
 - VIOS
 - alert 59
 - backupos 348, 725
 - cfgassist 420, 429, 590, 595
 - cfgdev 345, 491–492
 - chauth 463
 - chdev 499, 503, 600, 645–646
 - chkdev 183
 - chrole 463
 - chsp 574
 - chuser 450
 - cleargcl 451
 - cluster 571
 - entstat 596, 601
 - extendvg 425
 - fget_config 184
 - genfilt 431
 - help 421
 - installios 334–335
 - ioslevel 270, 272
 - ldapsearch 446
 - license 341

- loadopt 382
- lsauth 463
- lscfg 513
- lsdev 345, 349, 469, 483, 491–492, 500, 594, 645
- lsfailedlogin 451
- lsgcl 451–452
- lslv 172
- lsmmap 346, 350, 484, 508, 726
- lsnports 484
- lspv 467, 574, 646
- lsrole 463
- lssecattr 463
- lssp 574, 577
- lsuser 450
- mirrorios 426
- mkauth 463
- mkbdsp 347, 576
- mkkrb5clnt 448
- mklv 348
- mkrole 463
- mksp 347
- mktcpip 590, 595
- mkuser 445, 450
- mkvdev 492, 594, 599, 604, 725
- mkvg 348
- mkvopt 382
- mpio_get_config 468
- odmget 645
- oem_setup_env 451, 468, 645
- pcmpath query device 184, 468
- rmauth 463
- rmdev 346
- rmrole 464
- rmsecattr 464
- rmuser 450
- rolelist 464
- setkst 464
- setsecattr 464
- shutdown 451
- stopnetsvc 427
- su 450
- swrole 464
- tee 421
- topasrec 679
- tracepriv 464
- unloadopt 383
- vfcmap 484
- viosecur 615

- compressed pool 697
- concurrent volume groups 189
- configuration
 - virtual SCSI 49
- console 16
- continuous availability 100, 570
- control channel 591

D

- Dead Gateway Detection, DGD 237, 240
- dedicated adapters 263
- dedicated memory 20
- dedicated processor 112, 118, 135, 256
- dedicated processor donation 366
- dedicated-processor partition 118, 256
- defaultid 604
- DefaultPool 390
- defining the Virtual I/O Server partition 313
- defsysplanres command 721
- device drivers
 - Linux 74
- dirty memory pages 293
- DLPAR
 - See dynamic LPAR
- dm_multipath 499
- dmesg command 489
- DoS Hardening 432
- dynamic LPAR 17, 134, 256
 - IVM 260
 - RMC 260
- dynamic resources 256
 - considerations 117
 - micro-partitions 256
- dynamic routing protocols 237
- dyntrk 499, 503

E

- Electronic Service Agent, ESA 31
- Entitled Pool Capacity 390
- entstat command 596
- errorlog 515, 542
- errpt command 542
- ethchan_config command 608
- EtherChannel 65, 241, 265, 605
- EtherChannel Backup, ECB 603
 - configuration 604
 - testing 606
- Ethernet

- virtual Ethernet 64
- VLAN 16
- Ethernet connection bonding 570
- expansion command 445
- expansion pack
 - Virtual I/O Server 448
- extendvg command 425

F

- fast_fail 500, 503
- fast_io_fail_tmo 533
- fc_err_recov 499, 503
- fdisk command 498
- fget_config command 184
- file storage pool 39
- file-backed devices 174
- filemon command 294
- firewall
 - disable 429
 - setup 428
- firmware 16, 114, 117
- FTP port number 427

G

- genfilt command 431
- GNU Public License, GPL 141
- gsk7ikm command 432

H

- Hardware Management Console (HMC) 43, 726
 - editing virtual adapters 717
 - naming conventions 184
 - virtual device slot numbers 185
- Hardware Management Console, HMC
 - create virtual adapter
 - any client partition can connect 328
 - Ethernet adapter 325, 357
 - SCSI adapter 328, 357
 - virtual Fibre Channel adapter 478
 - dedicated processors 317
 - desired values 319
 - dynamic partitioning 588
 - enable connection monitoring 362
 - enable redundant error path reporting 331, 362
 - maximum values 319
 - maximum virtual adapters 324
 - minimum values 319

- mover service partition 315
- partition auto start 331
- shared processors 317
- use all resources in the system 316
- virtual console 53
- virtual storage management 471
- HBA and Virtual I/O Server failover 194
- hcall 17
- hcheck_interval parameter 726
- help command 421
- heterogeneous multipathing 196
- HMC
 - configuration 81
 - local 289
 - redundant 262
 - remote 288–289
 - RMC connection 265
 - roles
 - hmcsuperadmin 623, 625
- hmcsuperadmin role 623, 625
- Host bus adapter failover 193
- Host Ethernet Adapter, HEA 330, 360
- hscroot user role 625
- huge pages 264, 641
- hypervisor 293

I

- IBM i
 - alternate restart device 361
 - DS8000 Copy Services support 72
 - Ethernet line description 68
 - Ethernet redundancy 67
 - host partition 66
 - Independent Auxiliary Storage Pools 72
 - installation 374
 - load source 361
 - load source adapter 357
 - mirroring 73, 544
 - multipathing 73, 517
 - N_Port ID Virtualization, NPIV 71
 - network server storage space 66
 - PowerHA SystemMirror 72
 - queue depth 71
 - sector/page conversion 68
 - Shared Ethernet Adapter 67
 - storage system support 68
 - tagged I/O settings 361
 - virtual Ethernet 67

- Virtual IP Address, VIPA 67
- virtual LAN 67
- virtual SCSI 68
- virtual SCSI adapter 71
- virtual SCSI disk devices 377
- virtual SCSI disks 70
- virtual tape support 71
- IBM Installation Toolkit for Linux 378
- IBM Passport Advantage 139
- IBM Systems Director
 - VMControl 88
 - VMControl Enterprise Edition 90
 - VMControl Express Edition 89
 - VMControl Standard Edition 89
- IBM Tivoli Application Dependency Discovery Manager, TADDM 45
- IBM Tivoli Monitoring 46
- IBM Tivoli Security Compliance Manager 46
- IBM Tivoli Usage and Accounting Management, ITUAM 45
- IBM TotalStorage Productivity Center 45
- IEEE 802.1Q compatible adapter 326
- IEEE volume attribute 645
- inactive migration 264
 - capability 262
 - compatibility 262
 - example 102, 267
 - huge pages 264
 - migratability 263
 - multiple concurrent migrations 286
 - processor compatibility modes 292
 - remote 288
 - Shared Ethernet Adapter 266
- installios command 334–335, 342
- Integrated Virtualization Manager, IVM 44
 - virtual console 53
- inter-partition networking 61
 - virtual Ethernet 64
- interrupts 16
- ioslevel command 270, 272
- IP Address Takeover, IPAT 237
- IP fragmentation 231
- iSCSI 619
- IVE
 - LHEA 263

J

- jumbo frames

- virtual Ethernet 64

K

- kerberos 447
- kernel 2.6
 - Linux 74
- kickstart 379

L

- large pages, AIX 294
- layer-2 bridge 229, 231
- ldapsearch command 444, 446
- LHEA 263
- license command 341
- licensing
 - Linux 141
- Link Aggregation 236, 248, 265
 - Cisco EtherChannel 248
 - EtherChannel 248
 - IEEE 802.3ad 248
 - LACP 249
 - Link Aggregation Control Protocol 249
 - NIB 605
 - of virtual Ethernet adapters 250
- Linux 296
 - /sys/class directories 489
 - /var/log/messages file 489
 - automated installation 379
 - considerations 78
 - device drivers 74
 - distributor 141
 - Ethernet connection bonding 570
 - GPL 141
 - ibmvfc driver 489
 - kernel 2.6 74
 - LVM 78
 - mirroring 568
 - mpath 530
 - MPIO 75
 - multipathing 529
 - multipathing, MPIO 75
 - NPIV 533
 - RAID 77, 568
 - software licensing 141
 - VIO client 75
 - Virtual Console 75
 - virtual Ethernet 75
 - virtual Fibre Channel 75

- virtual media library installation 382
- virtual SCSI 75
- Yaboot 381
- Live Application Mobility 711
- Live Partition Mobility 187, 679
 - high availability 103
 - Multiple Shared-Processor Pools 126
 - remote 288
 - setup 618
 - validation 273
- Live Partition Mobility overview 80
- LMB 619, 624
- loadopt command 382
- logical HEA
 - See LHEA
- logical partition
 - Create Logical Partition wizard 314
- Logical partitioning technologies 17
- logical units 175
- logical volume 39, 171
- Logical Volume Manager
 - Linux 78
- logical volume storage pool 39
- logical volumes 173
- loose mode QoS 252
- LPAR workload group 263
- lparstat command 701
- lsauth command 463
- lscfg command 513
- lsdev command 345, 349, 469, 483, 491–492, 500, 594, 645
- lsfailedlogin command 451
- lsgcl command 451–452
- lshwres command 396, 654
- lslparmigr command 273, 286
- lslpp command 445
- lsmap command 346, 350, 484, 508, 726
- lsnim command 368
- lsnports command 484
- lspath command 511, 515
- lspv command 467, 511, 542, 646
- lsrole command 463
- lsscsi command 489, 498, 533
- lssecattr command 463
- lssp command 574, 577
- lssyscfg command 657
- lsuser command 450
- LVM mirroring 173
 - scenario 534

M

- MAC address 231
- Maximum Pool Capacity 390
- mdadm command 204, 570
- memory
 - affinity 294
 - available 625
 - LPAR memory size 293
- memory deficit 698
- Memory defragmentation 31
- memory expansion factor 696, 699
- micro-partitions 20
 - capped 113, 117, 256
 - configuration attributes 112
 - considerations 117
 - dedicated memory 20
 - I/O requirements 20
 - introduction 16
 - licensing
 - capped 136
 - entitlement capacity 136
 - uncapped 136
 - maximum 112
 - mode 113
 - processor entitlement 20
 - uncapped 113, 117, 256
 - uncapped weight 113
 - virtual console 16
 - virtual SCSI 16
 - VLAN 16
- migratability 263
 - huge pages 264
 - redundant error path 263
 - versus partition readiness 263
- migration
 - processor compatibility modes 292
 - remote 288
- migrpar command 287
- mirroring
 - AIX 540
 - IBM i 544
 - Linux 568
- mirrorios command 426, 451
- mirrorvg command 540
- mkauth command 463
- mkbdb command 347, 576
- mkkrb5clnt command 448
- mkldap command 445
- mkiv command 348

- mkrole command 463
- mksecdap command 439
- mksp command 347
- mksyscfg command 52, 192
- mksysplan command 723
- mktcpip command 590, 595
- mkuser command 445, 450
- mkvdev command 492, 594, 604, 725
- mkvg command 348
- mkvopt command 382
- Mover service partition 619
- mover service partition 262
- mpath 530
- MPIO
 - SDD 214
 - SDDPCM 214
- mpio_get_config command 468
- MSP 265
 - configuration 287
 - definition 82
 - network 293
 - performance 293
- multicast 229
- multipath command 499, 531
- Multipath I/O
 - Linux 75
- multipathing 493
 - AIX 511
 - IBM i 517
 - Linux 529
 - NPIV 533
 - user_friendly_names 530
- multipathing, MPIO
 - Linux 75
- Multiple Shared Processor Pools, MSPP 618
- Multiple Shared-Processor Pools, MSPP 21
- Multiple Shared-Processor Pools, MSPPs
 - architecture 22
 - capacity redistribution 122
 - capacity resolution 124
 - Level0 124
 - Level1 124–125, 388
 - capped 132
 - capping 130
 - ceded processor capacity 122
 - configuring 388
 - default Shared-Processor Pool 22, 124
 - deleting 126, 388
 - dynamic adjustment 127

- dynamic movement 389
- Entitled Pool Capacity 119
- examples
 - database 130
 - Web-facing 127
- HMC 388
- Live Partition Mobility 126
- maximum 388
- Maximum Pool Capacity 119, 126
- Reserved Pool Capacity 119

N

- N_Port ID Virtualization, NPIV 50, 678
 - adapter relationship 199
 - heterogeneous multipathing 196
 - IBM i support 71
 - NPIV 281, 533
 - NPIV-enabled SAN switch 191
 - redundancy considerations 197
 - SAN switch zoning 199
 - switch 283
 - using with IBM i 487
 - virtual Fibre Channel adapter 51
 - virtual WWPN 51, 190, 486
- N_Port ID Virtualization, NPIV
 - fast_io_fail_tmo 533
- NDP 229
- network
 - performance 293
 - requirements 82, 288
 - state transfer 293
- Network Installation Manager, NIM
 - AIX installation 368
 - bosinst.data script 335
 - master 368
 - repository 334
- Network Interface Backup, NIB 241
 - advantages 247
 - errorlog 607
 - EtherChannel 605
 - scenario 603
 - testing 606
- network security 427
- NIB 726
- no_reserve 506
- no_reserve parameter 724

O

- odmget command 645
- oem_setup_env command 451, 468, 645
- Open Firmware prompt 381
- operating system 16
 - AIX
 - boot image 688
 - SMIT 690
 - reboot 688, 692
- OS_install command 721
- OSI-Layer 2 226
- OSPF 237

P

- padmin
 - userid 345, 421
- paging space device 298
- parameter
 - hcheck_interval 726
 - no_reserve 724
 - single_path 724
- partition
 - auto start 331
 - boot mode 339
 - dedicated 256
 - DLPAR 134, 256
 - dynamic
 - IVM 260
 - enable redundant error path reporting 331
 - force shutdown 306
 - IBM i tagged I/O settings 361
 - inter-partition networking 61
 - licensing
 - capped 136
 - entitled capacity 136
 - uncapped 136
 - memory size 293
 - migrate 307
 - migration from single to dual VIOS 279
 - mirroring on two VIOS 273
 - multipath on two VIOS
 - MPIO 277
 - virtual Fibre Channel 284
- name 264
- processor sharing
 - allow when partition is active 367, 400
 - allow when partition is inactive 367, 400
- profile summary 332, 363

- readiness versus migratability 263
- recover 307
- redundant error path 263
- requirements 261
- resume 305, 667
- resume validation 665
- save current configuration 481
- shutdown 306
- standby/hibernated 82
- suspend 304, 660
- Suspend and Resume 82
- suspend capable 82, 655
- suspend validation 658
- workload group 263
- Partition Migration 315
- partition workload groups 636
- Passport Advantage 139
- pcmpath query device command 184, 468
- Peer to Peer Remote Copy 42
- performance 82
 - SMT 690
 - virtual SCSI 182
- Performance advisor 47
- performance monitoring 294
- Performance Toolbox 46
- persistent reservation support 61
- physical adapters 256, 263
 - requirements 262
- physical identifier 645
- physical processor 117
- physical shared-processor pool 116, 118
- physical to virtual migration 183
- Ping test 373
- pinned memory 294
- poold daemon 57
- port number 427
- Port Virtual LAN ID, PVID 231–232
- portCfgNPiVPort command 476
- portcfgshow command 476
- POWER Hypervisor 293
 - abstraction layer 16
 - dispatch cycle 129
 - firmware 17
 - inter-partition networking 64
 - introduction 15
 - partition integrity 16
 - virtual TTY console support 53
- POWER processor modes 681
- PowerHA SystemMirror 237

- IBM i 72
- PowerVM
 - activation 12
 - Enterprise Edition 11
 - Express Edition 9
 - features and technologies 4
 - Live Partition Mobility 11
 - Standard Edition 10
 - virtual SCSI 48
- ppc64_cpu command 692
- PR_exclusive reservation 61
- PR_shared reservation 61
- processing unit 112, 115, 117
- processor
 - assigned 134
 - available 627
 - binding 294
 - compatibility modes 292
 - dedicated 112, 118, 256
 - donation 366
 - physical 19, 117
 - POWER 16
 - unassigned 134
 - unit 112, 115
 - virtual 16, 114–115, 117, 256
 - virtual folding 116
 - virtual licensing 136

Q

- QPRCMLTTSK command 691
- Quality of Service 251
- queue_depth 496

R

- RAID 77, 211
- RAS tools 296
- readiness 263
 - battery power 263
 - infrastructure 263
- reboot 688, 692
- Redbooks website 731
 - Contact us xxvi
- redundant error path reporting 633
- remote login 427
- remote migration 288–289
 - infrastructure 290
 - private network 289
 - requirements 289

- workflow 289
- requirements
 - adapters 263
 - battery power 263
 - capability 262
 - compatibility 262
 - example 267
 - hardware 261
 - huge pages 264
 - memory 263
 - name 264
 - network 82, 287
 - partition 261
 - physical adapter 262
 - processors 263
 - redundant error path 263
 - Virtual I/O Server 262
 - workload group 263
- reserve_policy 218, 500
 - attributes 645
- Reserved Pool Capacity 390
- reserved storage device pool
 - creating 648
 - displaying 654
 - listing volumes 654
 - operations 301
 - suspend and resume
 - reserved storage device pool 298
- resource sets, AIX 294
- restricted Korn shell 421
- rmauth command 463
- RMC 234, 260, 262, 265, 288
 - connections 631
- rmdev command 346
- rmrole command 464
- rmsecattr command 464
- rmuser command 450
- Role Based Access Control, RBAC 678
- rolelist command 464
- rset command 116
- RTF310038003800300036003700 151

S

- SAN 263, 265, 270, 288
- SAN switch zoning 199
- SAP support 711
- save HMC profiles 726
- SCP 427

- SCSI mappings
 - defining 349, 470
- SCSI reservation 61, 619
- SDD 214
- SDDPCM 214, 468
- SEA
 - See Shared Ethernet Adapter
- SEA failover
 - load sharing 599
- secure copy (SCP) 427
- secure shell, ssh 427
- security
 - certifications 107
 - kerberos 447
 - LDAP
 - gsk7ikm 432
 - ldapsearch 444, 446
 - mkldap 445
 - mksecdap 439
 - network security 43
 - port numbers 427
- service port numbers 427
- setkst command 464
- setsecattr command 464
- shared dedicated capacity 23, 397
- Shared Ethernet Adapter 229, 266, 725
 - advantages 243
 - considerations 590
 - failover 65, 237, 591
 - control channel 239
 - delay 240
 - priorities 239
 - testing 595
 - GARP VLAN Registration Protocol 65
 - IBM i support 67
 - inter-partition networking 233
 - load sharing 245, 598
 - loose mode 252
 - quality of service 251
 - strict mode 252
 - TCP segmentation offload 65
- shared optical device 490
- Shared Storage Pools 39, 57, 677
 - adding nodes to a cluster 572
 - adding physical volume 573
 - adding physical volumes 573
 - checking the status of the cluster 573
 - cluster
 - create 571
 - repository 571–572
 - create logical units 572
 - creation 572
 - logical unit 56
 - logical units 175
 - map logical units 572
 - persistent reservation support 61
 - pre-requisites 219
 - thin provisioning 58
- Shared-Processor Pool 113
 - management using the command line 395
 - management using the HMC GUI 391
- shutdown command 451
- simultaneous multithreading 685–686
 - AIX 687
 - IBM i 691
 - instruction cache 686
 - Linux 692
 - mode setting 688
 - single-threaded execution mode 686
 - SMT-2 687
 - SMT-4 687
- single_path parameter 724
- smitty installios command 334
- SMS boot mode 339, 369
- SMS menu 340, 370
- smtctl command 688
- software licensing 131, 133
 - capped micro-partitions 136
 - CUoD 137, 139
 - dedicated processor 135
 - factors 134
 - IBM software 139
 - licensing factors 134
 - licensing methods 133
 - Linux 141
 - Multiple Shared-Processor Pools 133, 136
 - on demand 139
 - Passport Advantage 139
 - planning 139
 - shared dedicated partitions 135
 - sub-capacity program 139
 - enrollment 139
 - summary 137
 - uncapped micro-partition 136
- software maintenance 9
- software RAID 568
- SSA support 182
- SSH

- key authentication 289
- key generation 622
- SSH keys 725
- stale partitions 204
- startnetsvc command 447
- state transfer
 - network 293
- stopnetsvc command 427
- storage area network
 - See SAN
- storage pools 174
 - creation 347, 472
- storage virtualization 37
- strict mode QoS 252
- su command 450
- Suspend and Resume 82, 677
 - force shutdown 306
 - migrate 307
 - paging space device 298
 - partition 655
 - recover 307
 - resume 305, 667
 - shutdown 306
 - suspend 304, 660
 - validate 658, 665
- switch,virtual Ethernet 64
- SWMA 9
- swrole command 464
- syncvg command 542
- system
 - trace 294
- System firmware mirroring 30
- System Planning Tool, SPT 44, 713
 - Export the System Plan 724
- system WPARs 710

T

- target 48
 - device 349
- tcptr command 616
- tee command 421
- Thin Provisioning 677
- thread 117
- time-of-day clocks
 - synchronization 630
- Tivoli Identity Manager, TIM 45
- Tivoli Storage Manager, TSM 44
- Tivoli System Automation, TSA 237

- topas command 294
- topasrec command 679
- tprof command 294
- tracepriv command 464
- trunk
 - flag 238
 - priority 327
- TTY 53
- types of WPARs 709

U

- unassigned processor 134
- Uncapped 113
- uncapped 115, 117, 136, 256
 - weight 113, 117
- uncompressed pool 697
- unique identifier 645
- uniqueness
 - partition name 264
- unloadopt command 383
- unmirrorios command 451
- unused processor capacity 113
- user_friendly_names 530

V

- varyonvg command 204, 542
- vconfig command 75
- version command 475
- vfcmap command 484
- vio_daemon daemon
 - shared storage pool 57
- VIOS
 - See Virtual I/O Server
- viosecure command 615
- VIPA 237
- virtual adapter
 - slot numbering 273
- Virtual Console 53
 - Linux 75
- virtual disks 466
 - considerations for IBM i 348
 - file-backed devices 347
 - logical volumes 348
 - physical disks 345, 466–467
- virtual Ethernet
 - ARP 229
 - broadcast 229
 - Etherchannel 236

- flowchart 227
- IBM i 67
- IEEE VLAN header 226
- inter-partition networking 64
- introduction 61, 596
- jumbo frames 64
- layer-2 bridge 231
- link aggregation 236
- Linux 75
- multicast 229
- NDP 229
- POWER Hypervisor switch implementation 226
- PVID 231–232
- switch 64
- trunk adapter 226
- VLAN 233
- virtual Ethernet adapter
 - access external network flag 238, 326
 - creating a virtual Ethernet adapter for the VIOS 326
 - IEEE 802.1Q compatible adapter 326
 - maximum number 590
 - maximum number of VLANs 590
 - trunk flag 238
 - trunk priority 593
 - Use this adapter for Ethernet bridging 238
- virtual Fibre Channel 51, 270, 273, 281, 288
 - basic configuration 281
 - Linux 75
 - multipathing 284
 - requirements 283
- virtual Fibre Channel adapter 474
 - redundancy 193
- virtual I/O
 - client 74
 - IBM i 66
 - Linux 78
 - server 74
- virtual I/O adapters
 - Fibre Channel 51
 - TTY 53
 - virtual Ethernet adapter 223
 - virtual Fibre Channel adapter 192
 - virtual serial adapter 635
 - virtual TTY 53
- Virtual I/O Server 37, 265
 - access external network 326
 - as a LDAP client 432
 - command line interface 43
 - configuration 287
 - Denial of Service, DoS 615
 - Development engineer user 450
 - expansion pack 448
 - external storage subsystem 41
 - file storage pool 39
 - firewall 428
 - firewall disable 429
 - fix packs 344
 - HMC integration 43
 - installation 173, 184
 - installation media 333
 - latest enhancements 344
 - logical volume 39, 171
 - logical volume storage pool 39
 - managing users
 - creating users 450
 - global command log (gcl) 452
 - read-only account 452
 - service representative (SR) account 451
 - system administrator account 451
 - minimum disk storage capacity 336
 - mirroring 211
 - mksysb image 334
 - mover service partition 315
 - MPIO 213
 - multiple 272–273
 - network security 43
 - partition creation 313
 - Performance Toolbox support 46
 - planning 170
 - port numbers 427
 - redundancy 194
 - requirements 262
 - RMC 234
 - Role Based Access Control, RBAC 678
 - role-based access control, RBAC
 - authorizations 453
 - creating roles 463, 465
 - displaying a role's attributes 463, 465
 - displaying a user's role 465
 - linking a new user to a role 465
 - linking an existing user to a role 465
 - privileges 463
 - roles 461
 - SEA failover 272
 - Security Hardening Rules 431
 - service representative user 450
 - Shared Ethernet Adapters 37

- Shared Storage Pools 39
- single to dual 279
- storage virtualization 38
- supported commands 421
- supported platforms 38
- system administrator user 450
- system maintenance 102, 176
- System Planning Tool support 44
- third party applications 46
- third party support 41
- Tivoli support 44
- upgrading 176
- USB Blu ray support 678
- USB tape support 678
- using multiple 217
- using single 214
- version 270
- virtual media repository 39
- virtual LAN, VLAN 16, 233
 - IBM i support 67
- virtual media repository 39
- virtual optical devices 490
 - creation 491
- virtual processor 115, 256
 - licensing 136
 - reasonable settings 116
- virtual processor folding
 - default mode 116
- virtual SCSI 16, 270, 273, 288, 619
 - client adapter 48
 - client/server architecture 49
 - configuration 49
 - disk mapping options 184
 - error recovery 496
 - file-backed devices 174
 - heartbeat check interval 494
 - IBM i support 68
 - initiator 48
 - introduction 48
 - Linux 75
 - logical units 175
 - logical volume 171
 - logical volumes 171, 173
 - mappings 263
 - mirrored devices 209
 - number of devices per adapter 187
 - path timeout 496
 - performance considerations 182
 - server adapter 48

- shared optical device 490
 - slot number 188
 - SSA support 182
 - storage pools 174
 - target 48
- virtual SCSI adapter
 - IBM i support 71
- Virtual serial adapters 635
- virtual tape 679
 - IBM i support 71
- virtual TTY 53
- virtual WWPN 486
- VLAN 265
 - DLPAR 588
 - extending into client partition 586
 - HMC 588
- VMControl 88
 - Enterprise Edition 90
 - Express Edition 89
 - Standard Edition 89
- VNC 381

W

- workload group 263
- workload manager 294
- workload partitions 707
- WPAR 707
- WPAR characteristics 709
- WWPNs, world-wide port names 51, 190

X

- XRSET 294

Y

- Yaboot 381



Redbooks

IBM PowerVM Virtualization Introduction and Configuration

(1.5" spine)

1.5" <-> 1.998"

789 <-> 1051 pages



IBM PowerVM Virtualization Introduction and Configuration



**Understand
PowerVM features
and capabilities**

**Plan, implement, and
set up PowerVM
virtualization**

**Updated to include
new POWER7
technologies**

This IBM Redbooks publication provides an introduction to PowerVM virtualization technologies on Power System servers.

PowerVM is a combination of hardware, firmware, and software that provides CPU, network, and disk virtualization. These are the main virtualization technologies:

- ▶ POWER7, POWER6, and POWER5 hardware
- ▶ POWER Hypervisor
- ▶ Virtual I/O Server

Though the PowerVM brand includes partitioning, management software, and other offerings, this publication focuses on the virtualization technologies that are part of the PowerVM Standard and Enterprise Editions.

This publication is also designed to be an introduction guide for system administrators, providing instructions for these tasks:

- ▶ Configuration and creation of partitions and resources on the HMC
- ▶ Installation and configuration of the Virtual I/O Server
- ▶ Creation and installation of virtualized partitions
- ▶ Examples using AIX, IBM i, and Linux

This edition has been updated with the latest updates available and an improved content organization.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks