Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

BIM | the way we do it

Data Virtualization

How to get your Business Intelligence answers today

People matter, results count.

The challenge: building data warehouses takes time, but analytics are needed urgently

In the current dynamic business environment, data, both structured and unstructured, is growing exponentially. Organizations urgently need to utilize this data to make better-informed decisions.

The usual solution is to put in place mechanisms to extract, transform and load the data into an enterprise data warehouse (EDW) – a repository that provides a consistent and integrated view of the business, including its historical aspects.

Unfortunately, building an EDW, or even enhancing an existing one, is a laborious and time-consuming process that can typically take 6-24 months. Until it is complete, the business has no way to get an enterprise view of its data. Although there are scalable approaches to EDW (for example, more sources can be added to an existing EDW), there is no quick and robust way to make this happen.

The long EDW timescale is a concern for many organizations. Businesses want to improve their decision-making today, not in two years' time. Often, too, individual decision-makers want to see their reports and dashboards immediately, without waiting for the EDW process to complete, and so interim solutions are created. However, the underlying business intelligence requirements change frequently – 66% of them at least monthly, according to Forrester[1]. The likelihood is therefore that by the time the EDW is complete, the business's requirements will have moved on, and any analytics that have already been developed in early iterations will need to be reworked.

The solution: data virtualization

Data virtualization (DV) technology can solve this type of problem. DV techniques allow disparate sources to be combined within a logical layer or "virtual database" that can be accessed by reporting applications, downstream (i.e. consumer) applications, and Online Transaction Processing (OLTP) applications.

[1]Forrester Research, "Agile BI: Best Practices for Breaking through the BI Backlog", 2010.

The traditional way to report rapidly on data within transaction systems is either to download it to Microsoft Excel (with all the inherent problems of uncontrolled data and calculation errors), or else to build the reports directly on the source applications. These are both complex tasks, and the results may be inconsistent or incomplete owing to the use of different information hierarchies across source applications, and to the fact that source data may be stored at a lower granularity than the summarized level needed by reporting applications.

In contrast, DV maps the data from heterogeneous sources (both internal and external) and provides a virtualized layer on top of the source data that makes it understandable by target applications. This DV layer can achieve consistency by building a virtual data model on top of the source applications. This is a faster approach which requires less effort, because data is not really taken out of the source applications and so there is no need for physical extraction, persisting or massaging of data.

DV does not simply make source data accessible in a unified format. It can also carry out functions relating to data quality, data transformation, data masking, and so on, by applying transformation rules on the fly within the DV application.

The logical layer provided by the DV platform hides all the complexities from the downstream applications, and provides seamless access to data within disparate systems. This makes it possible to build analytical applications very quickly, because there is no need to spend time understanding the design of source systems or devising extraction methods.

DV can be deployed in a phased manner and can help organizations gradually build an enterprise data model. It is important to note that DV is not a replacement for EDW since one can't persist data using the DV platform. However, DV can certainly complement EDW by providing a virtualized layer on top of source systems (including the EDW itself, where available). This layer not only allows business users to generate reports quickly, but also provides a common, unified source of information that all reporting tools and applications can access – and does so in a matter of weeks.

DV is also useful for data cleansing because it facilitates user access to operational data. Users can look at a report or dashboard, drill into the data, identify where there is wrong data in the OLTP, or pinpoint an operational problem, all before the error gets onto someone else's radar. This is in stark contrast to a traditional data cleansing process that happens once a month or quarter.

DV has many other advantages. For example, a DV layer can allow reporting even when regulatory restrictions or departmental politics mean that data cannot leave its source. DV also facilitates usage of external sources like internet data

or public cloud-based applications for those building decision-making applications or enhancing existing OLTP applications.

DV also provides a convenient way for users to add their own personal data set and use it to analyze data provided by their company (for example, to introduce a new personal categorization or grouping).

DV in action

To understand the benefits of DV better, let's consider some practical applications in more detail.

Building a "single version of the truth" before the EDW is ready

DV is a quick and agile way to provide businesses with an enterprise landscape view so that they can take timely decisions. It can build a single version of the truth to which specific logical layers for specific subject areas can be progressively added.

At the same time, a DV initiative can pilot the EDW strategy. This can be achieved by:

- Building a logical layer on top of heterogeneous sources based on the business need
- Demonstrating this logical layer to the business
- Using feedback from the business to enhance the EDW strategy
- Gradually improving the logical layer by adding extra functionality and expanding the business scope
- Developing the EDW strategy in line with the improved logical laye

Enabling real time data access to support decision-making

To support their decision-making, businesses frequently need to integrate data from different source applications in real time. EDW technologies can do this in theory, but as we have seen, their long development cycle is often a source of frustration. In addition, the process of collecting data is not without an overhead: by the time incremental changes have been fetched, data quality rules applied, and the results loaded into the EDW, the data may no longer be current. Another option is to build a data store for operational reporting, but even this often cannot give users genuinely real time results because of data quality and transformation challenges.

DV technology enables true real time data access. The data continues to reside in the source database, so does not become out of date during the access process. Data from other business units, too, can often be accessed in its raw form without getting into a formal process of request and clarification, which means it, too, is available in real time. As a result, every part of the business can now access the up-to-date status of the business as a whole, instantly.

With the DV approach, it becomes possible to generate quick reports that, while not necessarily complex, provide powerful support for decision-making. For example, in the hospitality industry the reservation counter could find the revenue per available room (REVPAR) for a new customer based on a real time customer profile, and could combine this with current consumption patterns and social network sentiment analysis data.

An agile approach to business intelligence (BI)

With DV techniques, it is possible to build a virtualized layer for a small set of functionality in a matter of weeks. You can then create a few reports, get buy-in from the business, and progressively improve the virtual repository. In this way, DV can gradually build an enterprise data model.

This is an agile approach that achieves fast delivery, flexibility of architecture, and accurate results. The short development cycle facilitates quick wins. The business can focus on core functionality, instead of wasting time on resolving complex errors in non-essential data.

Meeting the need for a robust enterprise data bus

In a multi-application scenario, where data regularly moves from one application to another, organizations often look to build an enterprise data bus using technologies like JBossMQ or TIBCO. Another option is to build a physical data layer that pulls data from a variety of sources and makes it available to downstream applications in the format they require. This layer, once complete, can also provide an enterprise data bus.

Both these approaches have their place in systems integration, but DV offers a viable alternative where integration is primarily for reporting or read access, or where the level of application process integration is small compared with the reporting requirements. DV can provide an SOA-based solution, with consumer applications accessing the virtual layer through web services using SOAP, REST or WSDL.

The DV layer sits on top of the source system layer, and provides the target system layer with an integrated view.

How DV works

The DV technology includes three major components:

- **Integrated development environment (IDE):** User interface for development and access.
- **DV server environment:** Kernel of DV tools.
- **Management environment:** Monitoring, administration and error handling.

The DV server environment, which is at the heart of the DV technology, contains the following elements:

DV logical layer: Created by connecting the DV application to disparate databases, this layer represents an enterprise data model and provides a single version of the truth.

Query: Target applications can fire a query on source systems via the DV server.

Cache: The result set, once received, is stored in the cache database/file, so that it can be used in the future. If the same query is fired again by the target application, the query operates against the cached database, avoiding excessive hits to the source databases. This cache can be refreshed by a trigger or timer, or on demand. The DV server can store the cache in a standard database like Oracle, SQL Server, or DB2, or in any of the in-memory databases.

Query engine: Materializes the federated view of the data. Both cost-based and rule-based optimizers can be used to optimize queries.

Security layer: In addition to standard authorization and authentication mechanisms, the security layer provides row level security.

Objects: The primary objects created and used in data virtualization are views and data services. These objects encapsulate the logic necessary to access, federate, transform and abstract source data, and deliver the data to consumers. Object definitions change according to scope and requirement. Objects are normally created based on business functionality like supply chain, inventory, supplier etc.

Connectors: To connect to disparate data sources.

Metadata repository: Stores the connection details, data structures etc. DV servers can be clustered to achieve optimum cache performance.

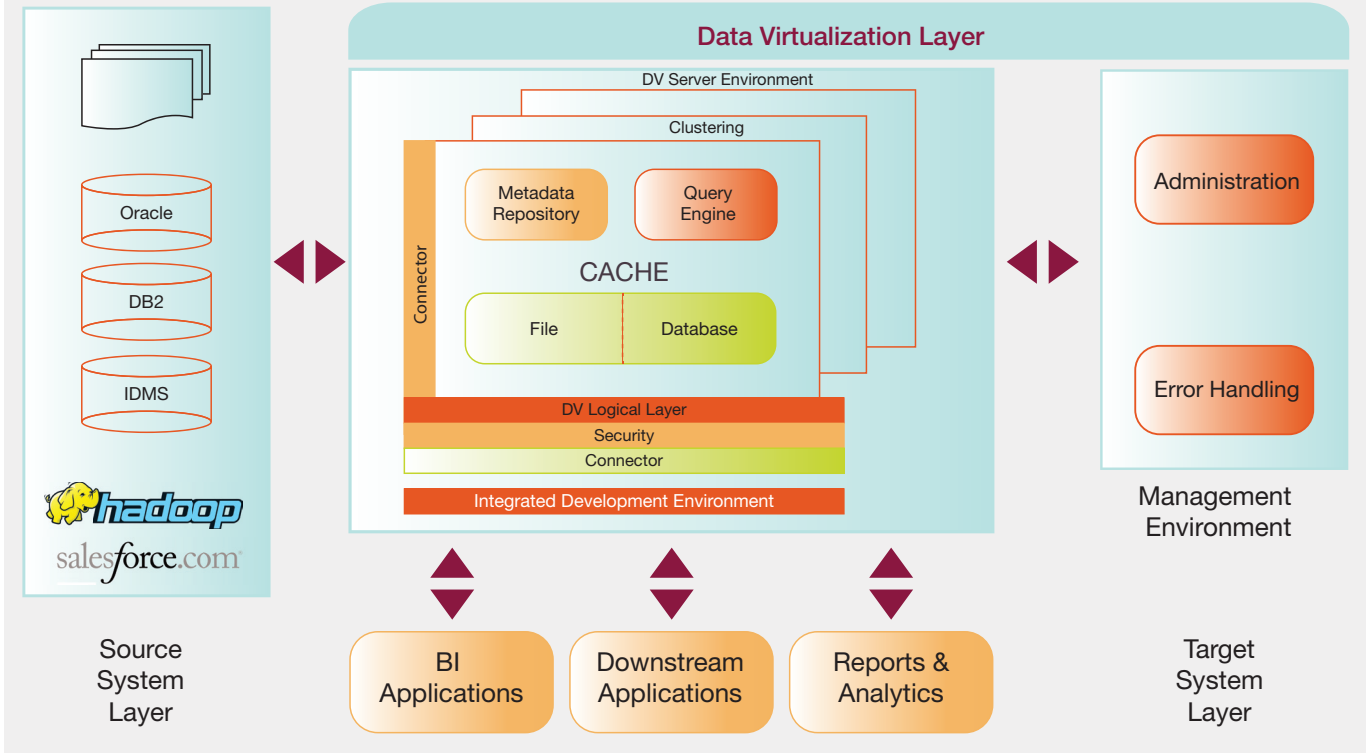
Business scenario: hospitality company

A hospitality company operates multiple casinos and luxury hotels. The casinos have both table games and slot machines. The hotels provide additional facilities such as spas, food and beverage (F&B), and booking for theater shows.

This company already has an EDW and associated analytical applications in place. These provide the following analytics:

- Consumption patterns of existing customers
- Performance analysis comparing tables with slot machines
- Comparison of this year's performance with previous two years

Figure 1. How DV works



- Expected revenue from each casino table in the next month(s)
- Room occupancy optimization
- Revenue generated by business unit (spas, F&B, rooms etc)

Useful as these analytics are, they do not provide what the company needs most: the capability to base decisions on real time data. That capability can be obtained by building a DV layer on top of existing source applications that capture and hold live operational data. The following real time analytics to support decision-making processes can then be based on the virtual layer:

- Amounts collected at a given time on all tables
- Suggested promotions for a hotel customer entering a spa
- Analysis of online purchases of show tickets, so that additional promotions can be offered immediately
- Promotional items that might attract a customer to table games after they have spent a certain amount on slot machines
- On-the-spot promotions that could tempt a customer who has lost a given amount in the casino to continue playing

Though these requirements could also be fulfilled by an EDW, there are inherent problems in doing so: the development lifecycle, the batch latencies that prevent truly real time processing, and the cost implications of additional data items in terms of storage and maintenance. For all these reasons, DV provides a superior solution for most real time requirements.

Success factors for DV

The success of virtualization technology in a given organization depends on the following characteristics of its source systems:

Stability: If the source systems/applications are not stable (e.g. the source system is stopped for six hours every night for back-up or maintenance, or the source system modeling often changes), any inheritance, even in the form of virtualization, will not give satisfactory results, and downstream applications will suffer.

Complexity: The more complex the source system is, the more complex the extraction logic has to be. You will need either to write complex queries or else to make the data more granular in order to make extraction easier. Either approach may impact the overall throughput provided by the DV layer.

Data quality: If the quality of data within a source system is below acceptable levels, it can be improved through the DV layer. However, this is a significant task and, again, can impact overall performance of the DV layer. Also, the extent of improvement that is feasible is limited: the DV layer can provide some validation, standardization and enrichment, but is not the proper place for in-depth data quality treatment. Given this limitation on improvement, the quality of the source data can influence the success of DV.

Granularity: Sometimes data within two different sources is maintained at different levels of granularity. For example, a company's supply chain application may store purchase orders at a daily granularity, while its finance application stores payment data at a monthly granularity. The difference in granularities means the data cannot be consolidated using the DV layer, so in our example it would be difficult to report what amount has been paid to a particular supplier against a purchase order.

In addition, certain principles must be adhered to in order to increase the chances of successful DV:

Observe legal constraints: Constraints on the use of data still apply even when the data is virtualized. For example, if certain data can only be used within Europe and other data within North America, that constraint will have to be observed when using DV.

Ensure consistency: If data from multiple systems is to be integrated, shared metadata is essential for ensuring consistency. Master Data Management (MDM) may be the solution and could, for example, be applied to customer, item, organizational and geographical content.

Avoid real-time updates: : The DV layer should not be used to update source systems in real time. Every application has its own framework for handling updates. If the DV layer is allowed to bypass that framework, the results can be unpredictable and inconsistent.

Be realistic about integration: DV is a good solution for real time and live data. However, it should not be used for historical or point-in-time reporting solutions which need extensive data integration in the form of application of business rules or transformations of the granular data. The reasons are that data cannot be persisted in the DV layer, and that applying data quality rules to historical data is a complex and potentially huge task.

The organization should review any proposed use of DV against these success factors. If the results of the review suggest that it is not possible to go ahead purely on the basis of a virtualized layer, it may be possible to adopt a hybrid approach, combining a virtualized layer with a physical one. For example, if there is already an EDW in place but it cannot provide data in real time, that requirement can be fulfilled with a DV layer, avoiding the need to enhance or rebuild the EDW. Pure reporting and analytics can continue to be done with the existing BI layer and EDW.

Performance Optimization

Many users fear that a new DV layer may create performance issues, because they feel that heavy query loads may choke up source OLTP systems. DV vendors now address this issue with an increased level of sophistication.

Most data virtualization tools persist or cache the source data in traditional databases or flat files. If they encounter a slow query, they can reconfigure the semantic layer as needed.

Many tool vendors are now enhancing their products to enable database caching to take advantage of in-memory databases – you can actually use an in-memory/columnar database as your database cache. DV solutions which embed an in-memory database can even be configured to replicate the source database in full, using mechanisms like Change Data Capture, in order to optimize both synchronization workload and data freshness. Once data is replicated in the in-memory cache, DV queries no longer affect the performance of source systems. This approach also provides optimized query execution, particularly for decision support type queries which require full data indexing in order to deliver speed-of-thought response time on any query. As in-memory appliances get more and more affordable, such an approach can provide an efficient and non-intrusive way to deliver high-performance reporting capability on a legacy system.

In addition, to prevent any unplanned performance impact on the source system, you can implement a resource allocation policy based on quotas. Most of today's databases allow you to set a quota so that the resources allocated to processing the queries generated by the DV layer are contained within predefined limits, which makes the impact of DV predictable.

Checklist: could DV help your organization?

- Is there mounting pressure from the business for quick, cost-effective reporting solutions?
- Is there a burning need to build real time reporting and analytics, and self-service BI?
- Do regulatory constraints stop you from replicating data, so that you need to access it within the source databases?
- Do you need to overcome departmental politics to allow your applications to share crucial information?
- Do you need to use an enterprise data bus to exchange data between applications?
- Do you need to integrate external sources (social networking, online sources, Dun & Bradstreet data etc) with your internal applications and BI?
- Do you need to combine multiple data types – for example unstructured or structured data from a public cloud application with internal structured data?
- Does your organization use multiple BI tools, each with a different presentation layer?

If the answer to any of these questions is "yes", then DV is worth investigating.

Conclusion

Data Virtualization is the best way to provide an interim solution for a single version of the truth with respect to multiple databases, until the time when an EDW is built. DV is a highly efficient way to make data from operational systems available for real-time needs, and it can also pull in older data from BI databases. That means BI tools can access all the company's business information, both current and historical, in a unified way.

Not only that: with DV it becomes possible for multiple BI tools to access the data in the same way, using the same business terminology and the same KPI aggregation rules. The organization can ensure everyone uses the same data, yet can still deploy a mixture of tools (whether reporting, dashboarding or predictive analysis) to suit different needs, locations and licensing arrangements.

The corollary is that there is no direct link, and so no technical dependency, between the databases and the BI tools. That makes it easier to upgrade, migrate, change or decommission a database or BI tool.

To sum up, the main benefits of DV are:

- A single version of the truth, even when multiple BI solutions are used.
- A huge time-to-market improvement: a "virtual data warehouse" can be available faster than any other solution.
- The ability to provide information in a controlled way, while combining historical and real time data.
- Mitigation of vendor and technology dependency by putting a neutral layer between databases and the applications that use them.

DV should be considered by any company that wants to address BI needs fast. Even for organizations that are also developing an EDW, DV can be used to supplement these developments and is a convenient way to model the requirements and confirm with the business that those requirements are correctly defined. That approach can significantly shorten delivery cycles for the EDW.

About Capgemini

With more than 120,000 people in 40 countries, Capgemini is one of the world's foremost providers of consulting, technology and outsourcing services. The Group reported 2011 global revenues of EUR 9.7 billion.

Together with its clients, Capgemini creates and delivers business and technology solutions that fit their needs and drive the results they want.

A deeply multicultural organization, Capgemini has developed its own way of working, the Collaborative Business Experience™, and draws on Rightshore®, its worldwide delivery model.

Learn more about us at

www.capgemini.com/BIM