

# Thinking outside the cube

Get the analytical flexibility and depth of MOLAP directly from the data warehouse. *by Alan Greenspan with contributions from Carlos Bouloy*

**B**usinesses today are complex, and they serve complex markets. So it follows that the decisions these organizations make and the actions they take are complex as well. To be competitive, retailers must make product mix decisions not across broad categories, but at the individual stock keeping unit (SKU) level, considering not just the department or subclass of products, but also the brand, size, color, variety and so on of each product within the department. At the same time, they must vary the mix by geographic region, division or store brand, location and customer demographics for each store.

Likewise, manufacturers make product introduction and production scheduling decisions based on a similar variety of factors, while telecommunications service providers devise service options by drilling into their customer data as opposed to simply analyzing their customer base. Within this context, critical decisions are made by viewing the data along many dimensions and from many angles.

The class of business intelligence (BI) that analyzes data dimensional in nature is called online analytical processing (OLAP). OLAP is implemented primarily through multi-dimensional OLAP (MOLAP) and relational OLAP (ROLAP). The key architectural difference between the two is that MOLAP utilizes a specialized pre-calculated data store, called a cube, while ROLAP utilizes a standard relational data store. Table 1, below, identifies the differences between the MOLAP and ROLAP methods of analyzing data.

## MOLAP-based analysis

To use MOLAP for dimensional analysis, the data to be analyzed must first be identified, then extracted from an enterprise data warehouse (EDW) and loaded into a staging area within the MOLAP system. Because the cube is a proprietary and specialized structure generally lacking scalability, the data needs to be summarized to meet the MOLAP tool's requirements.

MOLAP tools have excellent user interfaces

designed for "slicing and dicing" or rotating data on their various dimensions to allow flexible analysis. Useful analysis, however, requires accurate and up-to-date data. This necessitates the process of extracting, summarizing and loading to be repeated daily, weekly or monthly, depending on how volatile the data is, how often it is analyzed and how sensitive the analysis is to changes in the data. Apart from the data and analysis requirements, cube-build frequency is also often influenced by practical implications like time: many cubes require hours or even days to build. While it is generally viable to update cubes with appended new data on an ongoing basis, if the business structure (e.g., number of regions) changes or a product mix change affects history, a full cube build is required.

The MOLAP architecture consists of the EDW environment plus an external cube server accessed by the MOLAP tool's front end. Some MOLAP applications require a specific database source, necessitating an additional database layer, much like a single-purpose data mart. For example, Microsoft Analysis Services comes with and requires an SQL server database as the source for MOLAP cubes.

## ROLAP-based analysis

ROLAP uses a relational database as an alternative to a specialized cube as the foundation for OLAP dimensional analysis.

Specialized MOLAP cube structures were originally developed because relational database management systems (RDBMSs) couldn't provide the required performance for this workload. However, with the Teradata Database

**Table 1: Traditional MOLAP and ROLAP comparison**

	MOLAP	ROLAP
Data store	Specialized cube	Relational database
Flexibility	Single purpose store	Generalized multi-purpose store
OLAP query speed	Fastest	Slower
Analytical access	Specialized front-end tool	SQL
Dimensional analytical power	Highly specialized and powerful tool	Limited by relational database management systems (RDBMS) power

as the RDBMS, dimensional processing can be done in the EDW, removing the “requirement” to copy data from the EDW into the cube.

As with MOLAP dimensional analysis, preparing for dimensional analysis within the EDW starts with identifying the data to be analyzed. But because the analysis is executed against the relational tables and standard updates to the data are done directly in the EDW, no extracting, summarizing or staging is required. This enables the business analyst to take advantage of the sophisticated user interface of the OLAP tool while the IT organization configures the system to perform the actual data manipulation in a relational database.

To support dimensional analysis and provide high performance results, design changes are implemented. The steps include the construction of the Aggregate Join Indexes (AJIs) with pre-calculated information to accelerate OLAP queries. The AJIs are then maintained automatically by the database as the underlying data or staging tables are updated. (For an explanation of AJIs, see sidebar, pg. 59.)

The new complex queries of dimensional analysis changes the overall workload mix on the system. Use of Teradata Active System

Management should be implemented to ensure that service level agreements (SLAs) and organizational priorities continue to be met.

Until recently, MOLAP cubes were the only option for organizations seeking to do complex data analysis. However, as identified in table 2, below, ROLAP is emerging as a more cost-effective and flexible solution.

### Cube costs and limitations

With their user-friendly interfaces, MOLAP tools provide the flexibility to conduct detailed data analyses. For example, a retailer can see how a particular product at the SKU level sells in a particular store format within a specific region to a specified customer demographic. The retailer can then view the same data from a customer perspective to determine how many customers matching a particular demographic purchase specific SKUs within a product category based on store format within region.

Although the MOLAP tool interface and supporting specialized cube structure offer significant and unique capabilities, scalability is often a limitation. The cubes are typically limited in terms of the sheer data volume they can support. Within this, cube complexity (e.g., number

of dimensions in a single cube) is either physically limited or limited by the additional time required to load or build a more complex cube.

The technological constraints of MOLAP cubes limit the analysis that can be conducted. Consider that many organizations store more than three years’ worth of detailed data in their EDWs because they’ve found that trends established over this period can be valuable business predictors. But say that, for a particular company, a MOLAP cube can only hold two years’ worth of detailed information. In this case, the company must either summarize the data to get the full history loaded into the cube, or load the data as is and hope that it can gain valid insight through extrapolation.

In other cases, the number of natural dimensions in the data is greater than the MOLAP cube structure can accommodate. The reality with cubes is that compromises must be made in the analysis. These compromises potentially limit the questions that can be explored and, by extension, the resulting business value of the analysis.

Organizations must weigh several cost factors when evaluating MOLAP technology for dimensional analysis. The first is hardware and software licenses for the cube storage platform. MOLAP building and analysis is compute-intensive and will consume significant CPU resources on any server hosting the cube. Also, the cube is generally a very large representation of the detailed data with the various dimensions and intersections pre-calculated for high-performance analysis.

The second cost category is data movement. Keeping the cube data fresh for analysis requires regular maintenance. New extract, transform and load (ETL) processes must be designed and coded for every new cube. ETL scripts and processes must also be modified whenever changes occur to the EDW physical design for data represented in the cube or to the data elements required for a particular analytic process. These modifications require administrative attention, computing resources and maintenance. Cubes can answer many related questions, so they can often be reused, but new areas of analysis or subject areas require new cubes and ETL processes.

**Table 2: MOLAP vs. ROLAP cost comparison**

	MOLAP	ROLAP
Data storage	Large	Incremental
Extract, transform and load (ETL) process	Significant/external	INSERT SELECT SQL
Cube build	Periodic, processing intensive	Automatic Aggregate Join Index (AJI) maintenance
System hardware/software	Separate server and software license	Incremental storage and possibly processing capacity
Administration	Regular ETL and cube build, periodic ETL process update and maintenance	Data movement job execution (often automated), ongoing physical database design maintenance, workload management analysis and management

## Teradata Warehouse features supporting ROLAP today and into the future

With a Teradata Warehouse, the business analyst uses a dimensional analysis tool capable of using a relational data store. This is often the same tool that would be used in the MOLAP case. SQL is passed to the Teradata Database to perform the data manipulation and the results are returned. Aggregate Join Indexes (AJIs) are used in the database to pre-calculate many of the results, so very high performance is achieved as the front-end tool conceptually rotates the data for analysis.

As analytical requirements from the business continue to expand, Teradata Warehouse product capabilities continue to be enhanced to support those needs.

In Teradata Database V2R6.2, two features will enhance the capability to perform dimensional analysis in the data warehouse. Non-compressed join indexes can be partitioned using a partitioned primary index in the join index definition. This will provide even faster results and require less data to be processed on the database to satisfy dimensional queries.

The second new capability is the flexibility to define both triggers and join indexes on the same table. If intermediate tables aren't required for other design considerations, then the dimensional analysis AJIs can be defined on the base table, along with triggers used for event processing for active data warehouse applications. This simplifies the implementation of both analytical applications and active business applications on the same core detailed data.

Other changes are taking place in the dimensional analysis arena. Microsoft, for example, has established the multi-dimensional expressions (MDX) specification for dimensional queries. MDX provides a rich syntax for querying and manipulating the multi-dimensional data stored in OLAP cubes. Microsoft tools, including the ubiquitous Excel, use MDX to request dimensional analysis. By the end of 2006, the Teradata Database will provide support for MDX, enabling the capability to directly connect an MDX-enabled tool to the Teradata Warehouse using optimized SQL. —A.G. and C.B.

## Teradata Warehouse product features supporting in-warehouse dimensional analysis

Feature	Description/role
Aggregate Join Index (AJI)	Database-maintained structure with pre-calculated join and aggregate results allowing high performance query response; allows relational database management systems (RDBMS) to satisfy dimensional analysis demands without a physical cube structure
Referential Integrity	Database capability ensuring data consistency; when implemented on intermediate star tables, can help to ensure AJI has correct rows
TPump and Teradata Parallel Transporter Stream Operator	Data load utilities allowing data loads into tables with associated join indexes
Teradata Active System Management	Workload management suite; establishes workload groups and enforces rules so that dimensional analysis workload doesn't interfere with operational workload requiring tight service level agreements

System administration is the third major cost area. Maintaining cubes on a separate system requires administration. Cube development and life cycles must be managed, MOLAP software must be installed and maintained, and licenses must be managed.

## ROLAP cost and design considerations

Implementing ROLAP within the EDW is generally less costly than using a MOLAP-based solution. (Again, see table 2, pg. 58, for cost contrasts.) For one thing, the separate storage and processing platform for the cube—and any required intermediate staging storage—is replaced by incremental storage required within the data warehouse for join indexes and any other new tables. Due to the nature of specialized cube structures, the space required within the data warehouse is generally dramatically less than that required for the cube. For example, one customer saw a 55% reduction in total space required during a proof of concept comparing a Teradata ROLAP implementation with a Microsoft Analysis Services cube and SQL Server fact table implementation.

In addition, instead of MOLAP's ETL process and cube build, the ROLAP process features a SQL-based transformation and data movement process and automated AJI maintenance performed by the database itself. This not only saves significant administration and maintenance cost, but also helps to reduce batch windows and improve data availability.

Here are some design considerations for implementing in-warehouse dimensional analysis within a Teradata Warehouse:

- > Build a dimensional model for the cube solution defining all of the dimensions and their hierarchies that the cube will access. (See figure 1, pg. 60, for an example.)
- > Evaluate building the AJIs off of a star schema data mart built within the EDW. The star often closely matches the dimensional data and provides easy AJI definition and maintenance. Creating a physical star structure also separates the AJIs from the primary third normal form

Figure 1

## Sample dimensional model

Time	Product	Vendor	Location	Customer
Year	Time	Product	Vendor	Location
Quarter	Year	Dept	Parent Company	Region
Month	Quarter	Class	Company	Division
Week	Month	SKU		Area
Day	Week			Branch

Sample dimensional model defining the underlying data model a "cube" solution would need to access for multi-dimensional analysis.

tables and their load processes, thus separating the data warehouse load cycle from the dimensional model update cycle. The primary tables can be loaded with MultiLoad while the star tables with AJIs defined are periodically updated with an INSERT SELECT allowing control of when the dimensional analysis data is updated to support analysis requirements set by the business.

> Teradata Active System Management should be implemented to define and manage this new workload so that it receives the level of priority appropriate for analytical investigation versus customer facing or other operational workloads.

### A winner on all fronts

Dimensional analysis is a powerful business

decision-making tool in today's complex organizations and markets. Specialized tools have been developed to allow analysts to examine data on any dimension and to rotate the data to change the focus of the analysis at will. These tools have excellent analytical user interfaces, but in the past required specialized data stores for processing.

Through advances in relational database technology and features, this processing can now be performed within a Teradata Warehouse. Switching to this in-warehouse processing architecture saves money, reduces administrative effort, allows more flexible analysis on more detailed data and wider dimensions, and provides higher business value.

*Alan Greenspan is the product marketing manager for Teradata Database and load utilities. He has been with Teradata for 15 years.*

*Carlos Bouloy, a Teradata senior consultant currently in the Teradata Application Engineering organization, has been with Teradata for 15 years.*