

# Choosing the Correct OLAP Technology

**Author:** Tom Ferris  
**Organization:** Evaltech, Inc.  
Evaltech Research Group,  
Data Warehousing Practice.  
**Date:** 2/9/03  
**Email:** [erg@evaltech.com](mailto:erg@evaltech.com)



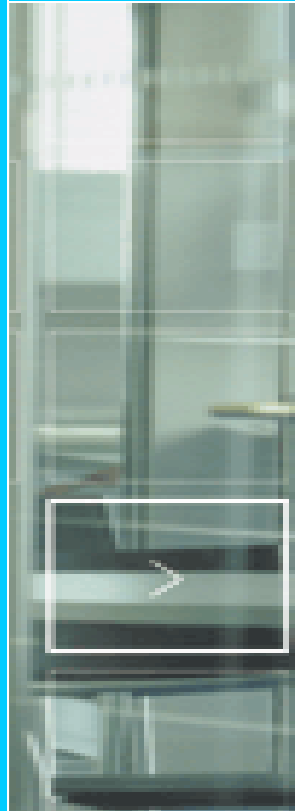
## Abstract:

Choosing the right architecture for your organization is the first step in making a technology work for you. Various alternatives are appropriate for specific application characteristics, but the industry is changing so rapidly that choices made today can be obsolete tomorrow.

Despite what you may have heard, Online Analytical Processing (OLAP) doesn't necessarily imply that you must use a multidimensional database as your decision-support server. The latest alternative, Relational/OLAP tools add decision-support capabilities to a relational database engine and are coming on strong.

## Intellectual Property / Copyright Material

All text and graphics found in this article are the property of the Evaltech, Inc. and cannot be used or duplicated without the express written permission of the corporation through the Office of Evaltech, Inc.



Overview .....	3
Why Decision Support is critical? .....	3
What is OLAP? .....	4
Multidimensional Databases (MDD) .....	4
Relational/OLAP Solutions .....	6
Criteria for Selecting Tools .....	6
Functionality .....	6
Fit .....	8
Performance .....	9
Scalability .....	9
Future .....	10
Summary .....	10

**Overview**

Choosing the right architecture for your organization is the first step in making a technology work for you. Various alternatives are appropriate for specific application characteristics, but the industry is changing so rapidly that choices made today can be obsolete tomorrow.

Despite what you may have heard, Online Analytical Processing (OLAP) doesn't necessarily imply that you must use a multidimensional database as your decision-support server. The latest alternative, Relational/OLAP tools add decision-support capabilities to a relational database engine and are coming on strong.

Most organizations are constrained by their ability to integrate and understand available, actionable business information. Decision support systems improve the decision-making process in an organization by providing timely, reliable information to decision makers. The best decision support systems include the necessary tools to analyze and visualize the correct information, and to provide a means to synthesize information as a result of the analysis that helps in the decision-making. The best of the best systems are architect to participate in a smooth and seamless flow of information, providing connectedness between all of the steps in the decision making process.

Decision support architecture is the broader set of requirements necessary to deliver the decision support system, including, but not limited to, data warehouses and OLAP (On-Line Analytical Processing) servers. The focus of this article is the complex process of choosing the best OLAP server for your organization.

Vendors of the OLAP products are innovating at an exciting pace. Any analysis of the features of each product is obsolete before it is complete. Instead, this article will describe the critical distinctions between classes of tools, divide the application space into some broad requirements and provide some guidelines for applying them in your own analysis.

**Why Decision Support is critical?**

Long on the periphery of IT practice, decision support is now the most talked-about topic in the press, conferences and the boardroom. The issue is critical because:

The future belongs to those who can see it and get there first. Decision support systems are a key element in understanding and dealing with the future. Despite two decades of work and untold billions of dollars of expense, operational systems developed with relational databases have failed to deliver on the promise of improved access to information. Though not necessarily a fault of the relational data model, current design methodologies simply lack the perspective to enable decision support, concentrating instead on reliability and transaction processing speed.

Data warehouses are not enough. Data warehousing is the hottest topic in corporate computing, but it only addresses a part of the decision support architecture. Choosing the right tools for OLAP provide the leverage needed to realize the benefits of data warehousing.

Get nimble. The current business environment is characterized by shortened product cycles, more volatile markets and an overall decline in traditional "Command and Control" organizations. As an alternative to traditional planning and budgeting systems, decision support provides a means to manage multiple plans in parallel, rapidly shifting

from one plan to another in response to changes. It provides instant results for "what if" and "what now" analysis.

Misinformation. The strengths and weaknesses of OLAP alternatives are poorly understood. The signal-to-noise ratio is weak, and the vendors aren't helping. There are clear application spaces for different products, but good information is hard to find.

### **What is OLAP?**

OLAP is often confused with decision support. At the practical level, OLAP always involves interactive querying of data, following a thread of analysis through multiple passes, such as "drill-down" into successively lower levels of detail. The information is "multidimensional," meaning that it can be visualized in grids. Information is typically displayed in cross-tabs, and tools provide the ability to pivot the axes of the cross-tabulation. These operations are always read-only. We call this activity "narrow OLAP" and distinguish it from "broad OLAP" (or just OLAP), which includes the following characteristics:

- Updating the database
- Modeling capabilities, including a calculation engine for deriving results and creating aggregations and consolidations
- Forecasting, trend analysis and other complex models, such as optimization, statistical analysis or other "esoteric" functions

Decision support encompasses all of the OLAP functions above, plus the end-to-end process of gathering, structuring, manipulating, storing, accessing, presenting and distributing actionable business information. OLAP is a significant subset of this activity. Regardless of the definition applied, OLAP is about enabling decision-making. Improved decision-making is our goal, not taxonomy.

### **Multidimensional Databases (MDD)**

From more than ten yards, it is difficult to differentiate one relational database from another. Granted, each has its own special charms, but they all conform to the same relational model, have roughly the same feature set and can be accessed by a common language, SQL. The exact opposite situation exists in the realm of MDD (multidimensional databases). The concept of a multidimensional database is actually rather simple. Rather than storing information as records, and records in tables, MDDs (logically) store data in arrays. Unfortunately, there is not much else that the different flavors of MDDs have in common. Each product is substantially different from any other. For example:

- Unlike the relational model, there is no agreed-upon multidimensional model
- MDDs have no standard access method (such as SQL) or API's
- Each product could realistically be put in its own category

The products range from narrow to broad in addressing the aspects of decision support. With those facts in mind, one can only evaluate different MDD in broad categories. At the low end, there are single-user or small-scale LAN-based tools for viewing multidimensional data. The functionality and usability of these tools is actually quite high, but they are limited in scale and lack broad OLAP features. Tools in this category include PowerPlay from Cognos, PaBLO from Andyne, Essbase from Hyperion, Business Objects from Business Objects, Microstrategy from Microstrategy. In fact, each

of these tools could realistically define a separate category; so diverse are their features and architectures. The pure multidimensional database engines are represented by Essbase from Hyperion, LightShip Server from D&B/Pilot and TM/1 from Sinper. Concentrating on just the multidimensional database capabilities of all of the products, there are two prevailing approaches to multidimensionality. The first, the hypercube, is exemplified by Essbase. In the hypercube model, symmetry is the paradigm. The "cube" in hypercube is actually a little misleading, since "cube" implies that each side is of equal length, which is rarely the case in these applications. The term hypercube is meant to describe a similar object of greater than three dimensions, also with flat sides and each dimension at right angles to all of the others. Having a hard time visualizing it? Everyone does. Despite the claims of "experts" in the industry that this is a more "natural" way to visualize data than a relational database.

Designing a hypercube model is a top-down process, with three major steps. First, you decide what aspect, or process of the business you will capture in the model, such as sales activity or claims processing. Next, identify the values that you want capture, such as sales amounts or elements of costs. This information is almost always numeric. Last, identify the granularity of the data, that is, the lowest level of detail at which you will capture it. These elements are the dimensions. Common dimensions are measure, time, scenario, geography, product and customer. For example, a single cell in a cube could refer to the budgeted \$Sales in January of this year, in the Northeast, of blue hair dye to K-mart.

MDDs are capable of providing stunning query performance, which is mostly a function of anticipating the manner in which data will be accessed. Because information in a MDD is stored in much coarser grain than a RDB, the index is much smaller and is usually resident in memory. Once the in-memory index is scanned (in a blink of the eye), a few pages are drawn from the database. Some tools are even designed to cache these pages in shared memory, further enhancing performance. Provided the application designer made the right assumptions about usage patterns, this scheme works pretty well. Another interesting aspect of MDDs is that information is stored in arrays. This means that values in the arrays can be updated without affecting the index. This is the reason that MDDs are so nicely suited for read-write applications. Unfortunately, many of them are pretty primitive in the way they handle multiple writers. Essbase is the exception in this case. A drawback of this "positional" architecture is that even minor changes in the dimensional structure require a complete reorganization of the database.

Another drawback in the hypercube model is that every value shares the same dimensionality. In the above example, we may use allowance % by customer, and it is redundant to store it repeatedly by geography (for example, the allowance % for K-mart is invariant across geography). Some hypercube products have techniques to get around this inconvenience, but the other multidimensional alternative, multicubes, uses a more elegant solution. Though implementations vary across products, multicubes dimension each variable separately and deal internally with the consequences. The downside is that these approaches are less straightforward and carry steeper learning curves. Conclusion: if your problem can be handled with a hypercube, you may be better off with one.

### **Relational/OLAP Solutions**

Though the name "Relational/OLAP or ROLAP" or, alternatively, "OLAP-on-Relational" is recently coined, the concept is not new. Pioneered by Metaphor, the concept of providing multidimensional analysis from relational databases is more than a decade old. The next-generation tools that trace their lineage from Metaphor include Information Advantage's AXSYS and Prodea's Beacon, companies founded by former Metaphor employees (Red Brick as well, though it is not, strictly speaking, an OLAP tool). Microstrategy's suite of DSS products and Stanford Technology Group's MetaCube, though not descended directly from the Metaphor ancestor, nonetheless owe their conceptual framework at least partly to it. Just like the MDDs, this is a rapidly evolving market, and all four of these products had their first production releases in the last year or two. Express, with improved SQL drill-thru to relational databases, slips into this group too. One could argue that, among its other capabilities, Holos provided ROLAP capabilities since 1988, or that Red Brick Warehouse, with any front-end that can produce or pass through RISO (Red Brick's extensions to SQL), is a ROLAP tool.

Consider a product a ROLAP tool only if it can meet the following criteria:

- Has a powerful SQL-generator, capable of creating multi-pass selects and/or correlated subqueries
- Is powerful enough to create non-trivial ranking, comparison and %-to-class calculations
- Generates SQL optimized for the target database, including SQL extensions
- Provides a mechanism to describe the model through metadata, and uses the metadata in real-time to construct queries
- Includes a mechanism to at least advise on the construction of summary tables for performance, preferably with the ability to monitor usage
- Ability to partition the application between clients, servers and a middle tier for managing threads to the database

### **Criteria for Selecting Tools**

The irony of ROLAP tools is that, for all of their sophistication, they are content with minimal client workstations, because the database server does most of the work. When we conduct an evaluation for our clients, we separate the criteria into five major categories: Functionality, Fit, Performance, Scalability and Future.

## **Functionality**

Matching your requirements to the functionality of an offering is an elusive prospect at best. If you listen carefully, you can almost hear the cells dividing in some of these products, they are evolving so fast! In addition, every day brings more press releases of vendors partnering with each other, linking their products to provide more end-to-end

functionality. In this atmosphere, it is often easier to start by identifying the show-stoppers and working backwards. Here are a few points to MDD as your solution:

- **Updating the database:** If your application calls for updating the database interactively, ROLAP is out of the question for now. First, the RDB schema used to get adequate performance, the so-called star schema, are remarkably inept for incremental update. Second, the SQL code generators are designed to generate only SELECT statements at this point.
- **Built-in features:** Most MDDs engines (Essbase in particular), as well as the high-end tools like Holos, Gentium and Acumate, have extensive libraries of financial functions, including currency conversion, depreciation, interest and Internal Rate of Return. ROLAP tools have just begun this process and are far behind. Other tools have quite sophisticated time intelligence, LightShip Server in particular.
- **Cross-dimensional calculations:** The phrase is a mouthful, but some examples include cost allocations and inter company eliminations. ROLAP is showing some signs of life here, especially DSS/Agent, so look closely for innovations. At the present, ROLAP can't handle it.
- **Row-level calculations:** Only rocket scientists can perform calculations down a result set in SQL, as opposed to across the columns. Simple metrics, like "margin = sales - cost of goods sold" can be performed in SQL by transposing the rows to columns, but there is a practical limit to the number of columns that can be handled with ease, even with multiple pass SELECTS. For applications that are like spreadsheets, such as an Income Statement, MDDs will be superior.
- **Rules rich:** ROLAP is not designed for modeling; most MDDs are. Acumate ES, Essbase, Express, Gentium and Holos have rich features for modeling, including the ability to develop user-defined functions, extending the capabilities of the product to suit your needs.

MDDs are subject to a few show stoppers, too. Here are some indications that a ROLAP solution may be best:

- **Data warehouse:** Data warehouses and relational databases are inseparable. If your requirement is to do OLAP analysis from a data warehouse, ROLAP is a natural; MDD doesn't make sense. However, subset-ing the data warehouse into smaller, manageable pieces, often referred to as data mart, is an area where MDDs hold some promise.
- **Rapidly changing dimensions:** Good examples of this situation are product codes that can be superseded at any time or customers who merge, change ownership or go bankrupt. At the detail level, this poses no threat to a MDD, but if the database carries historical data, the aggregations have to be run back to "the beginning of time." If this happens often enough, a MDD is the wrong choice.
- **Fluid Dimensionality:** MDDs excel when the dimensionality of a problem is neatly bounded and mostly static. Changes in the dimensional structure require a physical reorganization of the database, which is time consuming. Certain

applications are too fluid for this, and the on-the-fly dimensional view of a Relational/OLAP tool is the only appropriate choice.

- Data rich applications: In general, MDDs are designed to exploit the relationships between dimensional elements through their powerful calculation engines. Those applications with massive amounts of data (10's, 100's or 1000's of gigabytes) and relatively simple relationships are best left to the relational bases, for now.

Use MDD	Use ROLAP
Bounded Dimensionality	Fluid Dimensionality
Cross-Dimensional Calculations Row-Level Calculations	Multidimensional View of a Data Warehouse
Read/Write Applications Changing Dimensions	Rapidly Scale-Very Large Database, Many Users
Rules-Rich Applications	Data-Rich Applications
Data Marts	Data Warehouses

## Fit

Decision support doesn't happen in a vacuum. Unless the suite of tools can be rationalized in the computing environment of your organization, it will be very risky. The goodness-of-fit is dependent on a number of factors, such as:

- Development: Some tools require a fairly steep learning curve, particularly those at the high-end with the broadest functionality. Is your organization supportive of another sub-specialty, particularly if it is not similar to other initiatives? The 4GL languages of these tools are quite idiosyncratic, and the skills do not transfer well.
- Fat client/thin client: What is your existing IT architecture? Are you committed to providing state-of-the-art workstations to your OLAP audience, and the network bandwidth to serve it? If not, consider products that operate with a "thin client" and place a heavier burden on the server. Information Advantage, Prodea Beacon, Holos and Essbase all operate efficiently on midrange PC equipment. DSS/Agent, provided it can rely on DSS/Server, can as well. MetaCube and TM/1 (Perspectives, though the Spreadsheet Link moves this model to a server, which only transfers the burden to another platform) is more client-heavy, as is Express and Gentium.
- Network impact: Closely related, but not identical to the issue immediately above, is the impact placed on the network. Clearly, pure adhoc query tools without governors are the worst. Following closely behind are tools that assemble a multidimensional model in client workstation memory, like TM/1 Perspectives and, to some extent, MetaCube. Information Advantage will create local print

files and spreadsheet extracts on the server and broadcast them across the WAN, a less than optimal situation. Holos and Prodea Beacon can create a huge amount of network traffic between the application server and the database server. Shared caches between processes could alleviate this somewhat, but they are not implemented.

- **Connectedness:** MetaCube, for example, is completely OLE2 compliant. If your computing environment is committed to this form of distributed objects, MetaCube will make sense. Information Advantage, on the other hand, is the polar opposite and has a more UNIX/CORBA flavor. Does your organization have a history with MDDs, perhaps from prehistoric times with products like System W from Comshare or Express (same product, to a certain extent, has been around since the 70's)? Or is your organization committed to relational databases? Missionary work is probably not in your job description.

## Performance

Comparisons of speed between MDD and ROLAP are misleading. MDD advocates crow that there is no comparison between the almost instantaneous responses in MDDs, as opposed to the often-sluggish performance of a RDB. In addition to the fact that MDD applications performance poorly, consider this:

- The "comparisons" generally pit a standard OLTP design to a MDD optimized for OLAP analysis. Tuning a RDB with a "pure" star schema design, with precalculated and aggregated tables, along with a SQL generator that knows how to take advantage of them, is going to perform similarly to a MDD. If the database is Red Brick, the comparison is even more dramatic.
- The two approaches occupy separate application spaces that should rarely overlap. For that reason, speed comparisons are not terribly useful.
- MDDs have more subtle performance limitations. Queries that cross all arrays, as opposed to reporting from a handful of them, are similar in response time to full table scans in an RDB.

## Scalability

Since ROLAP tools rely on the RDB for servicing their queries, scalability is usually a function of the underlying database. There are some subtleties here, though. Without a middle tier to manage the interaction between client and server, it is possible to overwhelm the database server with the volume of separate processes. Advanced ROLAP tools provide a middle tier to "multiplex" the clients into a smaller number of active processes on the server. This technique can even serve to maximize the use of caching and shared memory by sharing processes. Also, complex OLAP queries often require multiple SELECT statements. The facility to submit these asynchronously as separate threads through the database server can have a huge impact on performance.

The actual size of the database is critical. Though many OLAP servers can scale up to multi-gigabyte size, careful analysis of "boundary conditions" (those situations that represent discontinuous reliability or worse, catastrophic failure) is necessary. For

11

example, what factors determine when an index can no longer fit in memory? Will the system be able to update the database in its bulk load mode within the update window each night or each month?

RDBs can handle much larger databases and take advantage of more powerful server architectures, but this is a distinct advantage for ROLAP TODAY. In a few months, it will start to dissipate. Here is the key point: RDB's have no advantage over MDD's architecturally for exploiting SMP and MPP, they've just been doing it longer (and not much longer). MDDs will catch up quickly. I think the MDD vendors are surprised by their own success and never dreamed they would be entering the 100GB+ space so soon.

## Future

Keep in mind that whatever tool(s) you select, you will have to live with your choice for a while. In that sense, it is just as important to evaluate the vendor as it is to evaluate the product. Will the vendor be able to keep up? What is the overall quality of the product? Is the vendor closely tied to a standard that is at variance with your direction? What kind of partner are you looking for? These are certainly "soft" issues, but important ones.

### Summary

Many factors muddy the water. We are building functionality for our clients in ROLAP that would be extremely difficult in MDD, such as comparable sales reporting or "on promotion" analysis. These applications are certainly multidimensional, but they require access to the lowest level of detail each time the query is run, and the power of a RDB and SQL is evident. Likewise, we've used MDD for applications that would have been impossible in ROLAP, like financial modeling. Here's another thing to consider: vendors are linking their products at a dizzying pace. You can no longer think about the features of a product in isolation, you must consider its "extended value chain" and how its partners complement it. For example, none of the ROLAP tools can stand on their own, they require a RDB. If you consider DSS/Agent or MetaCube or IA, you have to look at in concert with Oracle/Sybase/Informix or even Red Brick. Red Brick is the same, it lacks a front end. Same with Essbase. For example, Cognos' PowerPlay may be a poor choice in a given situation because it lacks the capability to store vast amounts of data and support dozens of simultaneous clients. Essbase may flunk the evaluation because the look and feel of its supplied front-end (spreadsheet add-in) is unacceptable for the intended purpose. But the combination of the two products (and they do work very well together) could be a perfect fit.

On the other hand, certain other tools are less cooperative and either offers no API or a limited one, like Express, Holos, and Gentium. Here the analysis is a little easier, because the products are more-or-less self-contained and you can decide if it has what you need. One thing is certain: vendors in both categories, Multidimensional Databases and Relational/OLAP are highly creative, capable and competitive. This subject area is finally getting the attention it deserves. With the exception of Express, now owned by Oracle, and Cognos, all of these companies are relatively small. But many of them are experiencing spectacular growth. This will attract more attention, capital and competition. All of us will benefit from it.

11

-----  
\*All the products mentioned in this article are the trade marks of their respective vendors.