OLAP

Adrienne H. Slaughter

Introduction

Product Table

Feature Table

A successful company today has many decisions to make. The better those decisions are made, the more successful, and profitable, the company is. To many chief decision makers, the ability to analyze faster and better than the competition means better decisions, higher profitability, and more success. The optimization of the relational database (RDB) has enabled companies to efficiently collect data about transactions, giving decision makers more information to use. However, there is an upper limit to the amount of data that one can have in an RDB and still perform an efficient analysis on.

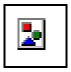
On-Line Analytical Processing (OLAP) allows users to perform quick and effective analysis on large amounts of data. The data are stored in a multi-dimensional fashion that more closely models real business data. OLAP also allows users to access summary data faster and easier. They can then drill down into the summary figures to get more detailed data, if need be. For a more detailed explanation of the OLAP rules, refer to Appendix A.

This paper describes the current state of the art of OLAP technologies. Some of the issues discussed are data storage, OLAP architectures, and what type of business structure will benefit the most from a particular OLAP feature. Leaders in the OLAP field are analyzed, and a substantial summary of the product and/or product line is given. Additionally, issues related to data-mining and data-warehouses are addressed.

OLAP System Components

An OLAP system is comprised of multiple components. A top-level view of the system includes a data source, an OLAP server, and a client. The data source is the source of data to be analyzed. Data from the source are transferred or copied into the OLAP server, where it is organized and prepared to provide short query times. The client is the user interface to the OLAP server. In this section, the function of each component and the significance in the overall system is described.

FIGURE 1. OLAP System Components



Sources

The source in an OLAP system is the server that supplies the data to be analyzed. Depending on the use of the OLAP product, the source could be an data warehouse, a legacy database housing corporate data, a collection of spreadsheets that holds financial data, or a combination of any of the above. The ability of an OLAP product to work with data from a number of sources is very important. Requiring that all source data is stored in a particular format or in a certain database is inconvenient for database administrators. It also reduces the power and flexibility of the OLAP product. Administrators and users find that OLAP products that allow data extraction from not only a wide variety of sources, but multiple sources, are more flexible and useful than those that have greater requirements.

Server

The back-end of an OLAP system is the OLAP server. This is what does all of the work (depending on the model of the system), and where data that is actively accessed is stored. Different philosophies govern the architecture of the server. In particular, a major feature of an OLAP product is whether the server uses a multi-dimensional database (MDDB) to store the data, or a relational database (RDB). This section describes pros and cons to each approach.

MOLAP

MOLAP stands for Multidimensional On-Line Analytical Processing. This means that the server uses an MDDB to store data. Because most OLAP products are based on an MDDB, the term OLAP usually refers to MOLAP as well.

The purpose for using an MDDB is fairly straightforward. It can efficiently store data that are by nature multidimensional, providing a means of fast querying of the database. Data are transferred from a data source (as described above) into the multidimensional database, and then the database is aggregated. This precalculation is what allows OLAP queries to be faster, since the calculation of summary data is already done. The query time becomes a function solely of the time required to access one piece of data, as opposed to the time to access many pieces of data and performing the calculaton. The approach also supports the philosophy of doing the work once, and using the results over and over [9].

Multidimensional databases are a relatively new technology. The use of MDDBs carry the same drawbacks that most new technologies do. Namely, they are not as robust as RDBs, and are not as optimized to the same extent. Another drawback is that most multidimensional databases are unable to be used while aggregating data, so it usually takes time for new information to become available for analysis.

Database Explosion:

Database explosion is a phenomenon of multidimensional databases. Though it is a problem that is experienced, it is difficult to explain how and why it happens. It appears to be related to the sparsity of the database and pre-aggregating the data. If a multidimensional database contains a small number of data points compared to the number of aggregation levels it performs, each piece of data will have a greater contribution to all data derived from it. When the database "explodes", the size of the database becomes magnitudes greater than it should be [2].

It is difficult to determine conditions for database explosion, or to predict whether a particular configuration will explode. One approach that does seem to help the problem is dynamic sparse data handling [12]. Dynamic sparse data handling allows a database to analyze it own storage patterns and optimize them to prevent database explosion.

TABLE 1. Multidimensional Databases

Pros	Cons
• Accurately models business data	• Generally does not handle VLDB's gracefully
• Fast access times with no SQL	• New technology that is not yet optimized
• Pre-calculated summary data	• Risk of database explosion

ROLAP

ROLAP is Relational On-Line Analytical Processing. The term ROLAP specifies that the OLAP server is based on a relational database. The source data are entered into a relational database, generally in a star or snowflake schema, which aids in fast retrieval times. The server provides a multidimensional model of the data, via optimize d SQL queries.

There are a number of reasons to choose a relational database for storage as opposed to a multidimensional database. RDBs are a well-established technology that has had plenty of opportunities for optimization. Real world use has led to a more robust product. Additionally, RDBs support larger amounts of data than MDDBs do. They are designed for large amounts of data [10]. A major argument against RDBs is that querying a large database with SQL to obtain summary data usually resulted in complex queries. An unskilled SQL programmer could easily tie up valuable system resources attempting to perform a query that is very simple in a MDDB.

TABLE 2. Relational Databses

Pros	Cons

• Ideal for large amounts of data	• SQL is not optimal for complex queries
• Proven, optimized technology	• Determining an optimal data-storage scheme is more important and difficult

Aggregated/ Pre-Aggregated Data:

Fast query times are imperative for OLAP. This is one of the basic tenets for OLAP-- the ability to intuitively manipulate data requires quick retrieval of information. Generally, the more calculation that needs to be done in order to produce a piece of information, the slower the response time. Thus, in order to keep query times small, pieces of information that are typically accessed frequently, but need to be calculated, are pre-aggregated. That is, they are calculated and then stored in the database as new data. An example of a type of data that may be precalculated is summary data, such as sales numbers for months, quarters, or years, when the data that was actually entered was daily sales numbers [PIL].

Different vendors have different approaches as to what values need to be pre-aggregated, and how many values to precalculate. Approaches to aggregation affects both the size of the database and the response time of queries. If more values are precalculated, a user is more likely to request a value that has already been calculated, thus the response time will be faster because the value does not need to be requested. However, if all possible values are precalculated, not only will the size of the database be unmanageable, but the time it takes to aggregate will be long. Additionally, when values are added to the database, or perhaps changed, that information again needs to be propagated through the precalculated values that depend on the new data. Thus, updating the database can be time-consuming if too many values are pre-calculated for the database. Since it is common for a database to be off-line while aggregating, it is desired to keep aggregation times small.

Client

The client is what is used to view and manipulate data in the database. A client can be as simple as a spreadsheet that incorporates OLAP features such as pivoting and drilling, it can be a specialized yet still simple report viewer, or it can be as complicated as a custom-built application designed for more complicated data manipulation. The World Wide Web is the newest form of client. This also bears the mark of a new technology; many web solutions vary greatly in the features included and the functionality of the solution as an OLAP solution. The different features of each client type are discussed in this section.

While the server is what provides the backbone of an OLAP solution, the front end is also very important. The server may provide a solid foundation for easy data manipulation, but if the front end is complicated or lacking in features, the user will not benefit to the same extent. The client is so important that many vendors focus on building only the front-end. What is included in these applications is a standard look and feel to the interface, pre-defined functions and structure, and a quick solution to more or less standard situations. For instance, financial packages are popular. A pre-built financial application will allow many to use common financial tools without having to design the structure of the database or common forms and reports.

Query Tool/Report Writer

A query tool or report writer provides a simple tool to access OLAP data. They have an easy-to-use graphical interface, and allow users to create reports by dragging and dropping objects into the report. While a traditional report writer will enable a user to quickly produce formatted reports, report writers that support OLAP produce live reports. The end product is a report that allows viewers to drill down to detail, pivot reports, support hierarchies, etc.

Spreadsheet Add-In

Many business currently do various forms of analysis on corporate data using a spreadsheet. In some ways, this is an ideal means of creating reports and viewing data. Analysts can create macros that treat the data in the desired way, a template can be designed so that when data is input, formulas calculate the correct values, and simple calculations don't have to be entered over and over. However, this provides a "flat" report, meaning that once a report is created, it is difficult to view it from different aspects. For instance, if a graph shows information for a time period, say a month. If someone wants to see the values per day (as opposed to the sum for the month), then an entire new graph needs to be created. New data sets need to be defined, new labels have to be added to the graph, and many simple yet time-consuming changes have to be made. There are also many areas where mistakes can be made, making it much more prone to errors. When OLAP functionality is added to the spreadsheet, one graph can be created, then it can be manipulated so that the user can see the information he or she needs, without the hassle of creating all possible different views.

The World-Wide Web as a Client:

The newest member in the family of OLAP clients is the Web. There are a great number of advantages to deploying OLAP reports via the Web. Themost significant is that no special software needs to be installed for someone to access the information. This saves lots of time and money for any organization.

Every Web product is different. Some make it easier to create Web pages, but are less flexible. Others allow you to create data views, then save them as static HTML files. These will allow someone to view the data on the Web, but they aren't able to actively manipulate the data. Others are interactive and dynamic, which allow them to be fully functional. Users can drill down, pivot, limit dimensions, etc. Before choosing a means of

Web deployment, it is important to decide what functionality one needs from a Web solution, and then determine which product will fulfill that functionality best.

Applications

Applications are a type of client that use OLAP databases. They are similar to the query tool or report writers discussed above, but they also incorporate greater functionality into the product. An application is usually more robust than a query tool.

Development

Typically, OLAP vendors provide a development environment for users to create their own customized applications. The development environment is generally a graphical interface that supports object oriented development of applications. In addition, most vendors provide an API that can be used to integrate an OLAP database with other applications.

OLAP Architectures

MOLAP

The multidimensional database market segment refers to companies whose primary concern are database engines. They are not necessarily concerned with what tools a user will use to use the database (i.e., the front end), but they provide a high-performance back-end so that the customer can create their own custom applications, or buy pre-built applicatons from another vendor. However, there some companies overlap in the MDDB/Application OLAP areas by providing both a front-end and a back-end.

ROLAP

The Relational-OLAP market segment is the smallest. Vendors in this area create an OLAP product that uses a relational database to store the multi-dimensional cubes.

Application OLAP

This is by far the largest area, and is generally what is thought of or referred to by the term OLAP. Application OLAP usually consists of a multidimensional database that is to be accessed by a particular application, or perhaps multiple applications. Vendors in this area primarily provide clients for the database. The client may simply be a viewer, or it can be a robust application that provides the user more.

Desktop OLAP

Desktop OLAP are products that do not necessarily connect to a server. They can essentially be run on a client, though they might download source data, in the form of a datacube, from a server. The fact that the datacube is built and stored on a user's machine makes it a good choice for those who may be frequently using portable computers, or who do not regularly perform such complex queries that a faster server is preferred to the speed of the client.

Vendor and Product Discussion

Different vendors implement OLAP based on their corporate philosophy of what the best OLAP product should include. The vendors discussed below take a variety of approaches, including those based on a multidimensional database, a relational database and applications that incorporate OLAP functionality on a variety of levels.

While an understanding of the different theoretical approaches to OLAP is important, actual performance is a quantitative measure. For this, The OLAP Council has developed a benchmark, the APB-1, to quantitatively compare the performance of various OLAP products. The OLAP Council is a consortium that was formed by a small number of OLAP vendors to support key OLAP tenets. To summarize, the tenets recognize that OLAP is a key technology for providing corporate analysts the tools they need for desired analytic capabilities. In April of 1996, the first OLAP-specific benchmark (APB-1) was released. [8]

The benchmark specifies parameters of a database and a set of 10 queries representative of normal business use. The OLAP vendor creates a compliant database and then runs the queries. A single measure, the AQT (Average Query Time), is generated based on the time it takes to load the database, aggregate the data, and then run the queries. Rules define what is and is not legal, such as whether to precalculate data values, which calculated data values can be stored, etc. [8]

While APB-1 results can be useful in determining whether or not a product can perform, they are not an ideal measure to base overall product usability. The benchmark is only run by vendors who build OLAP servers, as the benchmark is only relevant to the server. To qualify an APB-1 benchmark, the vendor must publish a report stating the specifics of the benchmark run. This includes the hardware, version information of the OLAP server and client, the Operating System, etc. This information should be used to determine the validity of an AQT (the result of the benchmark). For instance, while the AQT may be outstanding, the hardware may be unrealizable in a particular business setting.

In discussing the vendors below, benchmark information and pricing information is noted where possible.

MOLAP

Oracle

Oracle is the industry leader with Express, having a 20.7% share in 1997 [6]. Oracle acquired Express from IRI Software in 1995. The Express product line exhibits Oracle's enterprise approach to OLAP, including a server, front end, ROLAP capabilities, applications and web solutions.

[ORCL]

Server:

There are two Oracle OLAP server products. Express Server is the main OLAP engine for Oracle. It provides a multidimensional database, and is the engine behind other Oracle OLAP products. Additionally, Oracle provides Personal Express, which is an Express server run locally. It enables users to access and use a database when not connected to the server. This makes it ideal for mobile computing.

One high point for Express Server is the variety of data sources it is capable of using. Data for the multidimensional database can be gathered from a relational database (with help from Express Relational Access Manager), a multi-dimensional database, a spreadsheet file, or from a flat file.

APB-1: At 0.1 data density: Total time for load 10:43 (minutes:seconds), Total time for 250,000 queries: 20:15. Total elapsed time: 30:58. (March 18, 1998, [ORC1])

Administration:

Administering and maintaining an Express database is done by using Express's SPL, or Stored Procedure Language. Procedures are provided to define dimensions, data extraction routines, database organization, and various other tasks.

Development:

Users can develop custom OLAP applications for Express by using Express Objects. Objects provides a graphical interface for developers, and enables an object-oriented approach to building applications.

Express also provides an API to allow developers access Express databases through a variety of manners.

Client:

Oracle provides a multitude of ways to access the database. Express Analyzer provides a graphical interface to the database, and allows the user to easily produce reports. Analyzer can also share objects with Express Objects, as well as publish applications developed with Objects. Discoverer is the newest front end, and can be described more precisely as a data-query tool. It is more light-weight than Analyzer, but a favorite in the data query tool field. Express also has a spreadsheet add-in that can be used with Excel.

Applications:

There are two ready-made applications that can be purchased from Oracle. They are Financial Analyzer and Sales Analyzer. The use the engine and data cache of the Express Server, and they include report configurations tailored to the needs of sales and financial analysts. They are also helpful to the user because they define helpful functions that are commonly used in financial and sales analysis, yet need to be implemented in Express Server.

Web:

Web Agent and Web Publisher are the two web-authoring tools provided by Express. By using either of these tools, a user can create a web site that makes full use of OLAP capabilities by being both dynamic and interactive. Web Agent is more of a developers tool; it is a set of predefined procedures that are included in the Express Server SPL. Web sites can be built just as custom interfaces to a database can be built. For the end user, there is Web Publisher. Web Publisher is included with Express Analyzer, and enables those who have less programming expertise to create their own interactive web sites. Web Publisher is more like a wizard that guides one through the building of a web site, and provides a graphical interface to aide in the creation. It can be customized, though not to the extent that Web Agent allows.

Arbor

Arbor Software Corporation's main competitor is Oracle. The product is Essbase, and the most recent release is Essbase Server 5. Arbor is a popular backend to many OLAP products, meaning that many of the OLAP vendors and products out there now are don't necessarily produce the entire application, but may use Arbor (or another product) as a database, and then will produce an interface to the database. One example is Comshare, and the Web-component of Arbor, CrystalInfo, is produced by Seagate Software. Recently, Arbor has announced a partnership with IBM to provide IBM with an OLAP solution. The multidimensional data store in Arbor's Essbase Server will be replaced by IBM's DB2. This appears to be a ROLAP system, but in fact it isn't. It is just an OLAP system with out the best of the multidimensional database good points, but with high points of a RDB system. That is, you may not get the best query time, but more data will be able to be stored, RDB's structure will be utilized, and SQL will be the language used to access the data. This is useful when data is to be shared with other SQL-based systems.

[ARB]

Server:

Arbor Essbase 5.0 is the newest release of Arbor's OLAP server.

Arbor implements a variety of security features, allowing administrators to specify permissions down to the cell level. Additionally, users can be required to provide a password for either read or read/write access to particular data. The passwords can be specified to expire after a set amount of time, or after a period of user inactivity. These features provide administrators a means to implement flexible security policies for the database.

APB-1: AQT is 0.0113 seconds/query. (Mar 11, 1998 [ARB1])

Essbase is supported on the following platform:

Win95

WinNT

OS/2

AS/400

HP-UX

IBM-AIX

Sun Solaris

Tools:

In addition to the Essbase Spreadsheet Add-in that gives users the power of OLAP in their spreadsheet, Arbor provides WIRED for OLAP (an analysis and presentation tool), Crystal Info for Essbase (a report writer and scheduling system), and SQL Drill-Through, which enables users to view detail data in source relational databases.

Applications:

Arbor has recently released the Arbor Essbase Adjustment Module. This is an application that aides users in the preparation of reports that need to be produced on a regular schedule. It helps to automate report formats and accounting processes. There is also the Arbor Currency Conversion Module, which is able to convert multiple currencies to a common currency, based on a model that is defined to account for exchange rates.

Development:

Essbase Objects is a powerful tool that developers can use to develop applications for Essbase. It incorporates an object-oriented approach to OLAP programming, using a comprehensive set of 32-bit ActiveX controls. One stand-out feature of Essbase Objects is that it can compiled into optimized Essbase code, allowing it to perform better. Essbase Extended Spreadsheet Toolkit also enables developers to build custom apps that extend spreadsheet functionality. Last but not least, Arbor also publishes an API that allows developers to incorporate Essbase databases into programs in C/C++ and Visual Basic.

Server Management:

Essbase Application Manager provides a GUI to Essbase databases and applications. It allows easy maintenance and administration of databases, objects, and applications. There is also a SQL Interface, which allows users the ability to access data from a relational database and use it in the Essbase Server.

Essbase Servers from $37,500.

Applix

Applix is a company headquartered in Westboro, MA. They produce a unique product in that the OLAP server does not pre-aggregate the data or batch-process data. Instead, they implement something they refer to as "Real-Time Analytical Processing" or (RAP). Only explicitly entered data is saved in the multidimensional cube; this keeps the size of the cube down. When a user performs a query, the server responds by either retrieving or calculating the value. The calculation is performed only on demand.

RAP has its advantages and disadvantages. One advantage is that the threat of exploding databases is non-existent with a MDDB that is not aggregated. Only values that are explicitly requested are computed. This seems to be contradictory to the premise of MDDB's. After all, a major building block of the MDDB idea is that if values are aggregated and pre-computed, then retrieval time will be minimal. TM-1 performs the calculations in RAM however, which allows it to respond to a query quickly. One reason this approach works is because just as MDDB's are generally sparse (which causes database explosion), data retrieval is also sparse. That means that processing overhead is saved because unaccessed values are never computed.

Another feature added to the latest release of TM-1 is the ability of the server to store a view that takes a long time to calculate. If a particular view takes N seconds to compute (an arbitrary time, but longer than acceptable), that view will be stored in the database for future accesses to that view. Applix estimates that approximately 10% of the total size of the database will be accounted for in stored views. Future releases hope to further improve performance by extending the saved-view concept to individual cubes, as opposed to the entire database.

[APP]

Server:

TM-1 Server 6.0 is the newest release of the Applix server. It is actually an OLAP server that implements RAM-based calculations. Additionally, TM-1 Perspectives is a single-user OLAP server coupled with the TM-1 Client to provide a personal OLAP database while working off-line.

Platforms:

Windows 95

Windows NT

Sun Solaris

HP

RS6000

APB-1: AQT is 0.03842 seconds/query. (Audit performed Nov. 1997 [APP1])

Client:

TM-1 Client is a spreadsheet-based interface with OLAP functionality. However, many companies provide front-ends for TM-1, such as Comshare and arcplan, Inc., or integrate TM-1 into business solutions for a more focused industry, such as Creeth, Richmond and Associates, Inc. Thus, TM-1 powers many applications that require OLAP functionality.

Development:

The TM-1 API and TM-1 ShowBusiness provide development support for custom OLAP applications. TM-1 ShowBusiness is an object-oriented graphical development interface. It incorporates presentation graphics, reporting tools, and hypertext authoring capabilities. Additionally, LotusNotes can be used to help share information in the TM-1 database.

ROLAP

MicroStrategy

MicroStrategy is the industry leader in ROLAP. Their philosophy is that there should be no constraints on the size of the data warehouse, so there are no questions about growth. Because they are a relational OLAP producer, DSS scales extremely well. DSS server is a right step for a company to take when moving towards a data warehouse.

MicroStrategy doesn't have an OLAP engine that can be run locally, which is inconvenient for users who are frequently running laptops or for users who simply want to work off-line. However, they do have a product, DSS Broadcaster, which allows data to be sent to a variety of output devices. This product currently stands alone, though other companies are taking steps towards producing something along these lines.

DSS Broadcaster sends data upon request or when a particular event happens. For instance, a daily update can be sent to a manager reporting the amount of revenue for the previous day. This information can be sent via email, to a pager, a mobile phone, or a fax.

Server:

DSS Server is at the center of the DSS product line. It is a powerful engine that enables other agents to access a relational database in a multidimensional manner. DSS Server incorporates a variety of database drivers that allow it to optimize to the desired relational database (they support Oracle, DB2, Sybase, Red Brick, Informix, and other relational databases). Additionally, emphasis is placed on the ability of the database to scale upwards, and includes a driver to adapt for Very Large Databases (VLDBs), up past the terabyte level.

The nature of a relational OLAP product (namely that it requires a relational database) restricts MicroStrategy from being able to provide a true personal server for work off-line, but through DSS Agent (see Client) a subset of data can be downloaded and analysis performed off-line.

Client:

DSS Agent is the front end or client tool for DSS Server. One outstanding feature of DSS Agent is the use of intelligent agents to automate business processes. For instance, with DSS Agent, an agent can be created that will know when and where to look for data, and then will know what to do with it (that is, how to clean it and where to put it). With the use of agents, many mundane but frequently repeated tasks can be automated.

DSS Executive utilizes the power of DSS Agent to provide a higher-level report writer and analysis tool. It uses an object-oriented approach and a drag-and-drop interface to allow users to quickly produce executive information system applications.

Last but not least, MicroStrategy provides an Excel Add-In that can be used to give OLAP functionality to a spreadsheet.

Development:

DSS Objects provides developers with the ability to develop applications integrating DSS Server. DSS Objects allows developers to use Visual Basic, Delphi, and C/C++ to access a DSS Server. DSS Objects comes packaged with an Excel Add-In that quickly ready to use.

Applications:

MicroStrategy doesn't have any applications or suites; the closest they get is DSS Broadcaster, which uses information from a database and can actively transmit that information through a variety of channels. Data can be sent on schedule, or it can be instigated by a change in data parameters (exception reporting).

Web:

DSS Web 5.0 is the new generation of web-enabled OLAP for MicroStrategy. One significant feature of DSS Web 5.0 is the support of Microsoft's webcasting standard. This allows web pages to automatically be sent to a users computer. Some of the more significant features of DSS Web include the abilty to save charts or graphs from the web, a report wizard, and custom report packages. DSS Web takes adavantage of Java and ActiveX to make a web page created in DSS Web to be customizable. Additionally, a security plug-in for Netscape provides a means of authenticating users prior to access.

Application OLAP

Comshare

Comshare is an example of an application OLAP product. That is, it's an application, but it incorporates OLAP functionality into the product.

Server:

Comshare Decision is flexible in the server that is used with it. Applix TM-1, Arbor Essbase, and Oracle Express are all supported multidimensional database servers that can be used with Decision. Other than features that are specific to the servers (such as TM-1's RAP), the choice of server is transparent.

Hyperion

Hyperion Software is a client-only OLAP producer. The newest product is Hyperion MBA, or Multidimensional Business Analyst, which replaces HyperionOLAP. HyperionOLAP was a popular, high-powered front end to Applix's TM-1 database. With the most current market information (that of 1997), Hyperion has the second largest market share, surpassed only by Oracle [6].

The nature of the product ensures that much of the functionality is based on the server database, namely TM-1. Hyperion is popular for the added intelligence that it provides in the form of smart dimensions, predefined functions, and reporting capabilities. The focus is to provide a robust financial package that incorporates OLAP.

Server:

HyperionMBA and Analytical Accountant use Applix's TM-1 server.

Platforms:

Windows NT

Client:

Hyperion provides two front-end OLAP solutions. The first, HyperionMBA, employs OLAP to enable business analysis. As is typical with application OLAP, Hyperion incorporates smart dimensions and pre-defined functions for accounting and currency manipulations. Hyperion's Analytic Accounting software incorporates OLAP features into an accounting package.

Desktop OLAP

Cognos

Cognos is a good example of a desktop OLAP product, which means that a majority of the processing is done locally, as opposed to on a server. Impromptu is a query tool that is used to colled data from a multidimensional database. The data is then put into PowerPlay, which holds a powercube on the desktop.

Incorporating OLAP with DataMining and Data Warehouses

Frequently, OLAP is mentioned in the company of datamining and data warehousing. The three have developed as the business world has begun to realize the value of data.

Relational databases were revolutional in that they allowed businesses to collect data from day-to-day transactoins on a large scale. Via SQL, it was possible to perform rudimentary analysis on the data. When more complicated analysis was desired, it was discovered that SQL and RDB's were not the ideal solution. Spreadsheets were able to provide more flexibility in analysis, but with drawbacks. Data needed to be imported to the spread sheet from a database, and a spreadsheet was not able to deal with large amounts of data efficiently.

Today, more and more companies are implementing data warehouses and using data mining tools for their data. Data warehouse provide a repository for cleansed corporate data. Transaction data is verified for correctness, staged, and then stored in the warehouse. Data mining tools allow corporate analysts to discover hidden trends in data. While it may appear that an OLAP system provides both a data warehouse and data mining tools, there are subtle differences.

An OLAP tool simply provides a way to perform quick and easy analysis of data. Generally, the user-analyst has an idea of what to look for in a view. They want to have a means of manipulating the data to help visualize patterns in the data.

A data mining tool will attempt to discover hidden trends in data. It is not a concern to be able to provide quick and easy view of data, as it is the tool, not the user, that is performing the analysis. Typically, a data mining tool looks for trends and patterns that are difficult for a human to comprehend from a large amount of data.

Summary and Conclusion

This report discussed the various aspects to be considered when determining which OLAP product will best fulfill a user's needs. An overview of considerations that are important when analyzing an OLAP product included the pros and cons of different implementation decisions. Additionally, products that were exemplary of these features were introduced and discussed.

To summarize, these are points that need to be considered:

- Where is the data coming from?

  The data to be analyzed can be located in a variety of places. Perhaps the OLAP database will be accessing data from the corporate data warehouse or OLTP system. Sources discussed earlier were spreadsheet files, ASCII files, and relational databases. If the OLAP product is already capable of accessing the data source, less data staging and cleaning procedures need to be implemented.

- What is the user doing with the data?

  Once a user has accessed the database and begun performing analysis, it is important that the user be able to deal with the data accordingly. Depending on the needs of the user, it may be preferable to have a robust report writer, or the ability to produce and publish dynamic web pages may be desired. Alternately, it may be preferable to the user to be able to create custom applications quickly and easily.

- How much data is there?

  This is a major factor in determining an OLAP database. Relational OLAP products are able to handle large amounts data better than Multidimensional OLAP products. If the amount of data doesn't require a relational database though, a multidimensional product will be just as helpful.

- Who is the user?

  The level of expertise of the user is important when determining the front-end of an OLAP system. Some users will be more comfortable integrating OLAP with a spreadsheet, while others will prefer a specialized application. Related to the level of expertise of the user is whether or not training is to be provided. A large company may be willing to pay for user training, smaller companies may not be.

The front-end should be one that the users feel comfortable with, and are able to manipulate effectively.

Appendix A: OLAP Rules

The concept of OLAP and business intelligence was around long before OLAP was officially defined. The official birth of OLAP, however, was in 1993, when Arbor Software commisioned Dr. E.F. Codd to formally define the distinguishing characteristics of an OLAP product or system. Though the corporate nature of the definition of the OLAP rules implies that they can't be accredited to Codd ubiquitously, there are few arguments to the validity and solidity of the 12 original rules. After the original rules were defined, more were added by other companies, namely IRI Software (which was bought by Oracle).

OLAP evolved from spreadsheets. Business analysts use spreadsheets such as Excel or Lotus 1-2-3 to create reports. Add-ins were used to extend spreadsheet capabilities when features such as the ability to rotate a table became desired. More and more features were added, such as drilling up and down in dimensions. Eventually, the extensions and the reasons they were added were formalized, and it was discovered that a more intuitive way for modeling data for analysis was possible.

The following listing of the OLAP rules are adapted from E.F. Codd's original paper [12]. Addendums to the rules are also included, adapted from The OLAP Report's "What is OLAP?" on-line paper [1].

Basic Features

The basic features are those that are the core of the theory behind OLAP and multidimensional analysis. There are two key concepts to keep in mind when learning about OLAP: First, an OLAP implementation should allow many users access to the same data in whatever way they want to. Also, OLAP is providing a user a way to get the most information out of the data, and OLAP uses the idea that a human is going to be able to get most out of the data if they are able to access it any way they want. The user should be able to follow their thoughts when looking at the data, and the OLAP product should be able to follow along.

Multidimensional Conceptual View

Multidimensionality is key to OLAP. Some may claim that an OLAP product is simply a multidimensional database. That's not quite true. OLAP products may be based on a multidimensional database, but the important part is the conceptual view. ROLAP products are able to store data in a relational database while providing a multidimensional view of the data.

Multidimensionality is important because of the multidimensional nature of business data. Common dimensions are time, geographical region, and product. An example of a data measure that can have this dimensionality it the number of products sold; another is the amount of money that is brought in (either during a time period, in a particular location, or for a product). Analysts may be interested in different numbers or reports. Thus, it makes sense to define the measures similarly, and redundancy in the database can be reduced.

A multidimensional view is intuitive for human users. As the number of dimensions grow, it is more and more difficult for humans to visualize the model of the database. However, the visualization of a three-dimensional "cube" can be generalized to more dimensions.

Intuitive Data Manipulation

While the multidimensional view aides in the storage of data and query times, intuitive data manipulation is intended to reinforce the ability of humans to perform successful analysis. It is very common for a person to look at an initial table or graph that is a summary of some data, see a value that might not look quite right, and then want to learn more about what goes into that value. When using a static reporting tool (either on paper or on-line), it is necessary to generate an entirely new report in order to see more detail. This entails determining where the data comes from, defining an area, and then putting it into a report format. Depending on how automated a system is, it might not be too difficult. With an OLAP tool, it can be as easy as a double-mouse click, and that value is be exploded into more detailed data. Another example is that sometimes understanding a table can be easier when it is rotated for some reason; an OLAP tool can easily change the axis of a table or graph to make them easier for the user to understand and gain knowledge from.

This rule is to disallow complicated procedures to change a view. The flow of analysis is interupted when, to change a view, a user must select the correct item form a menu, enter some information into a box, and then wait for the view to come up. Intuitive implies that there is a graphical interface and communication tools such as mouse clicks and drag/drop routines.

Accessibility

In practice, an analyst will need data from a variety of sources, including an OLTP database, a legacy system, or perhaps spreadsheet files. The OLAP server should be able to access these multiple sources seamlessly when queried.

The OLAP database will contain much of the data, however most systems also have a legacy database (an old database) that contains historical data that has not been stored in the production database (the newer, in this case OLAP, database). A summary of the historical data may be in the production database, but if a user does need to see the detailed historical data for some reason, the OLAP server should be able to provide that data without a flaw.

The example of a legacy and production database is a simple one. In reality, a company can have many, many data sources. Data can come from flat files, spreadsheets, active relational databases, or perhaps other multidimensional databases. An OLAP product must be able to deal with all of these different types of sources and provide that data to the user without a problem.

Batch vs. Interpretive Extraction

This rule addresses the idea of preaggregation of data and live access to data, as well as the accessibility to live data sources. Batch extraction refers to the capability of a product to gather data from various sources and stage it in the OLAP database. Interpretive extraction involves accessing the data from the various sources when it is required for a query. Few products offer a flexible choice between the two. Ideally, a product would allow a user to specify which data should be batch extracted and which should be accessed live. For instance, an analyst may want to have historical data available in the database, but then should be able to access newly generated data in an OLTP.

OLAP Analysis Models

OLAP is for analysis, thus it should support various different models for analysis. Examples of analysis models are static reports, reports that allowing paging, rotation, and drilling, what-if analyses, and models for budgeting and performance issues. Some OLAP products are by nature better for particular analysis models, though all products produce static reports and also allow other analysis models to some extent. Typically, OLAP vendors who focus on the front end or client are able to provide a greater number of analysis models, and the models tend to be more flexible as well.

Client Server Architecture

A client/server architecture is important for an OLAP product in the sense that it is the best way to deploy information to many users. In order for all analysts to produce coherent results, they should have access to the same data. It is also more efficient from the view of the database to have all of the data consolidated in one place.

Transparency

The requirement for transparency comes from the ability of OLAP products to have a variety of different clients. A client may be an application, a spreadsheet, or a web browser, and can be created either by the a user-analyst or by another vendor. Because of the wide variety of possible clients for a system, it is important that the OLAP product ensure all OLAP functionality to the user, however it is accessed. There should be no loss of functionality to a user if he or she uses a spreadsheet to access the data, or a custom-built application.

Multi-User Support

A robust OLAP product will allow multiple users access to data concurrently. Multi-user support also includes the ability of the server to allow individual users to use different analysis models.

Reporting Features

The reporting features are integral to a good data analysis tool. Some personal preference goes into the style of the reports that are created, but there are a couple of rules that reporting tools should follow in order to comply with full OLAP functionality.

Flexible Reporting

The ad-hoc nature of analysis required for OLAP products dictates that a reporting tool should be flexible. A user should be free to show a display of the cube in any way; that is, there should be no constraints on the arrangement of the dimensions on the axes of a table, and cubes, when displayed, should be able to be rotated and sliced and diced.

Uniform Reporting Performance

Again, to prevent a disruption of flow when performing analyses, the time it takes a tool to present a report should be consistent. This means that the queries should be run at a consistent pace, and also that the report is generated in consisted times as well. As far as the query is concerned, when the database is fully pre-calculated, there shouldn't be any problem with uniform query times due to the fact that there is just a retrieval time, and no calculation time.

Automatic Adjustment of Physical Level (Dynamic Handling of Sparse Data)

Every database has an optimal configuration, determined by the number of dimensions and the amount of data that it contains. For optimal query times and data storage, a database should be able to determine the storage of the data based on these factors.

Dimension Control

Generic Dimensionality

This rule is intended to stipulate that all dimensions are equivalent, and that an operation performed on one dimension will similarly perform on another dimension. A dimension can be given special rules or formulae, but those rules should be extensible to all other dimensions. This is a contreversial rule; many vendors want to make dimensions "smart" for ease of use by analysts. "Smart" dimensions ususally include time or perhaps special accounts. Time dimensions will understand the structure of a year into months, quarters, etc., and will treat values dimensioned by time accordingly. Another example of smart dimensions are account dimensions, such as expense accounts.

The extent to which a product adheres to this rule depends on the needs of the user. For some users, it is better that extra rules are already accounted for by the database. For users who need more flexiblity, or have little use for the dimensions that are specially defined, a more generic approach is favorable.

Unlimited Dimensions & Aggregation Levels

Ideally, an OLAP product should allow users to define as many dimensions as needed to fit their business model. Within each dimension, users should be able to determine the number of aggregations levels as well. While unlimited dimensionality is theoretically required, technically it is impossible. Statistically, it is determined that few business models will exceed 15 dimensions, 20 dimensions is a better number to target [12].

Unrestricted Cross-dimensional Operations

All forms of calculations must be allowed across all dimensions.

References

[ORCL]

Oracle Express Server Datasheet, Oracle Corporation. 1997.

Oracle OLAP Products: Adding Value to the Data Warehouse. Dave Menninger, Oracle. 1997.

Oracle OLAP Technology: Enabling Better Business Decisions.

[ORC1] Oracle(r) Express(r) Server Blazes to New OLAP Benchmark Record on Sun Servers. Press Release, Oracle Corporation. May 18, 1998.

[ARB]

The Essbase Product Family Datasheet, Arbor Software. 1997.

"Increasing Corporate Intelligence with the World Wide Web". Arbor Software.

[ARB1] Arbor Essbase APB-1 Audit Report. Arbor Software, 1998.

[APP]

TM-1 Datasheet, Applix. 1997.

[APP1] Applix TM-1 APB-1 Execution Audit Report. Applix. December, 1997.

[PIL]

An Introduction to OLAP: An explanation of multidimensional terminology and technology. Pilot Software.

[MS]

Relational OLAP: An Enterprise-Wide Data Delivery Architecture. A MicroStrategy White Paper.

The Case for Relational OLAP. A MicroStrategy White Paper.

Relational OLAP Server: DSS Server Datasheet. MicroStrategy.

[HYP]

HyperionOLAP Technical Data. Hyperion Software.

[1] The OLAP Report: www.olapreport.com.

[2] Pendse, Nigel. "Database Explosion". The OLAP Report, January 17, 1998.

[3] Pendse, Nigel. "Market Segment Analysis". The OLAP Report, January 18, 1998.

[4] Pendse, Nigel. "OLAP Applications". The OLAP Report, January 18, 1998.

[5] Pendse, Nigel. "OLAP Architectures". The OLAP Report, January 18, 1998.

[6] Pendse, Nigel. "Market Share Analysis". The OLAP Report,

[7] Pendse, Nigel. "What's in a name?". The OLAP Report,

[8] The OLAP Council: www.olapcouncil.org.

[9] Koutsoukis, Nikitas-Spiros, Gautam Mitra, Shirley de Jonk, and Cormac Lucas. On-Line Analytical Processing: The Interaction of Information and Decision Technologies. in press

[10] Saylor, Michael, Manish Acharya, and Robert TrenKamp. "True Relational OLAP: TheFuture of Decision Support." Database Journal, Nov-Dec 1995.

[11] Andrews, Dave. OLAP, MOLAP, Overlap. BYTE Magazine, August 96.

[12] Codd, E.F., S.B. Codd, and C.T. Salley, Providing OLAP (On-Line Analytical Processing) to User-Analysts: An IT Mandate. E.F. Codd & Associates, 1993.

On-Line Information:

Oracle OLAP: www.oracle.com/st/products/options/olap/

Arbor: www.arborsoft.com

MicroStrategy: www.strategy.com

Hyperion: www.hyperion.com

Gentia: www.gentia.com

Applix: www.applix.com

Comshare: www.comshare.com

Cognos: www.cognos.com

The OLAP Report: www.olapreport.com

The OLAP Council: www.olapcouncil.org