Architecture and Benefits of PowerOLAP™

White Paper

PARIS Technologies, Inc.

Architecture and Benefits of PowerOLAP™

PowerOLAP™ represents a milestone in the evolution of OLAP (on-line analytical processing) technology. Like any important evolutionary event, PowerOLAP combines the most advanced features of what came before it with new capabilities. Most significantly, PowerOLAP enables users to *reach through seamlessly to access transactional data in a relational database for dynamic OLAP manipulations in a true multidimensional environment*. In addition, PowerOLAP employs Excel as a front end, connecting users throughout an organization with underlying data sources via the tool they know best, the familiar spreadsheet, direct to their desktops.

This paper will examine the logic for the design of the PowerOLAP architecture. We will consider first, briefly, the necessity for OLAP in the organization and the development of the OLAP market; next, different OLAP product types, their strengths and shortcomings; and finally, in greater detail, how PowerOLAP supersedes predecessor products, the functionality that makes it unique, and the benefits that accrue to an organization that employs PowerOLAP as its OLAP tool.

OLAP in the Organization

The term OLAP, for *On-Line Analytical Processing*, is less than ten years old, but has been adopted to describe a variety decision support products. The individual who coined the term, E. F. Codd, is known as the father of the relational database. Codd noted that *relational database management system* (RDBMS) products do not have the functionality to perform sophisticated analysis on transactional data, the core operational data of a business. Relational systems were developed for capturing, validating, and storing large volumes of individual transactions: their strength *is on-line transaction processing* (OLTP). However, their static physical design—data is stored in two-dimensional tables, with each entry being a row in a table—stymies the fast, flexible modeling required for sophisticated business analysis.

The capability to perform sophisticated analysis—specifically, the multidimensional analysis provided by OLAP—is by now well-established as an organizational objective. Analysts need to view data along the multiple dimensions that define an organization: its products, regions where it operates, time-periods it organizes data by, measurements of business activity, the types of customers it sells its products to, versions (*Budget*, *Actual*, *Forecast*)—essentially, the number of dimensions necessary for the creation of effective business models.

Various products claim to provide OLAP functionality, which has led to questions over how OLAP should be defined, what product features an OLAP tool should include, what constitutes true OLAP functionality. *The OLAP Report*, an independent review of OLAP technology and products, has defined OLAP as “fast analysis of shared multidimensional information.” Until now, with the advent of PowerOLAP, OLAP

products so defined have tended to fall into two general categories (ROLAP and MDBs), discussed below. By examining the strengths and shortcomings of these product types, we can appreciate how PowerOLAP goes beyond ordinary OLAP—in effect, how PowerOLAP, by seamlessly integrating both relational and multidimensional worlds, along with the Excel spreadsheet, combines different products' strengths and overcomes their limitations, raising the bar for optimal OLAP functionality and performance.

OLAP Product Categories — ROLAP and MDBs

ROLAP

ROLAP products (for *relational* OLAP) directly access data stored in relational databases. Consequently, they can readily retrieve and update transaction data. ROLAP products also enable organizations to leverage their existing investments in RDBMS software. These are key strengths—which, we shall see, PowerOLAP shares.

ROLAP products access a relational database by using SQL (structured query language), the standard language that is used to define and manipulate data in an RDBMS. Subsequent processing may occur in the RDBMS or within a mid-tier server, which accepts requests from clients, translates them into SQL statements, and passes them on to the RDBMS.

ROLAP products have made strides in providing GUIs (graphical user interfaces—drag-and-drop, point-and-click tools) and in generating SQL execution plans that remove end-users from the SQL writing process. However, this over-reliance on processing via SQL statements—*including processing for multidimensional analysis*—is a serious drawback. Whether it is generated "transparently" or not, SQL is the language of relational tables: it isn't stretching the metaphor to say that SQL's vocabulary is too limited and its grammar too inflexible to accommodate the most sophisticated modeling required for multidimensional analyses.

There are further drawbacks to structuring a multidimensional model solely within relational tables: Before end-users can submit requests, the relevant dimension data must be extracted and reformatted in denormalized structures known as star schema or snowflakes (so-called because of the way the tables are conjoined). These tabular structures are necessary to provide acceptable analytical performance. Sophisticated ROLAP applications also require that aggregate tables be pre-built and maintained, eliminating the need to process summary data at runtime.

The necessity to reformat data and pre-build tables within the relational database in advance of the actual querying points up two issues:

1. Since it is unlikely that end-users have the skills to do this advance work, MIS professionals set the parameters for analytical processing.
2. Once a multidimensional model is created in a relational format, it becomes difficult to alter the model by then adding new optimization schemes. Consequently, the ability to alter/add dimensions for modeling is limited.

In summary, ROLAP products show deficiencies in providing fast and flexible modeling capabilities. In spite of this, they have become increasingly popular as an OLAP product choice because of their access to relational databases.

Multidimensional Databases — MDBs

Rather than provide a multidimensional view of relational data, as ROLAP products do, MDBs (also known as MDDBs or MOLAP products) enable end-users to model data in a multidimensional environment. This is a real product strength, as it provides for the fastest, most flexible method to process multidimensional requests.

The structure of a multidimensional model is not a series of tables (as exists in a relational database) but what is generally referred to as a *cube*. Cubes modeled in a multidimensional database extend the concept associated with spreadsheets: just as a cell in a spreadsheet represents the intersection of two dimensions (sales of product by region), a cell in an n-dimensional cube (*products*, *customers*, *region*, *months*,…*n*th dimension) represents the intersection of *n* dimension members. As in a spreadsheet, a cell might be calculated by formulas involving other cells.

In short, multidimensional databases allow users to add extra dimensions to the cube, rather than additional tables as in a relational model.

The cube structure allows for particularly fast, flexible data-modeling and calculations. For one, locating cells is vastly simplified—an application can identify a cell location by name (at the intersection of dimension members) rather than by searching an index or the entire model (via SQL SELECT statements), as in a relational database.

Further, multidimensional models incorporate advanced array-processing techniques and algorithms for managing data and calculations. As a result, multidimensional databases can store data very efficiently and process calculations in a fraction of the time required of relational-based products.

What, then, are the drawbacks of multidimensional databases?

1. Relevant data must be transferred from relational systems—a potentially "redundant" re-creation of data in another, albeit multidimensional, database. Once data has been transferred, there is no simple means for updating the MDB as individual transactions are recorded by the RDBMS.
2. MDBs have limited storage capabilities—generally no more than 1 gigabyte.
3. MDBs are proprietary systems—unlike the relational model, there is no agreed-upon multidimensional model, nor any standard access method (such as SQL) or APIs.

In summary, MDBs' product shortcomings correspond to ROLAP products' strengths—although MDBs provide the fastest means for developing flexible multidimensional models, their chief drawback is their *lack of access* to transactional data stored in relational stores.

Now, in the context of the ROLAP/MDB dichotomy, we can situate PowerOLAP and explain how it supersedes these two predecessor OLAP products.

PowerOLAP—Modeling in an Optimal Multidimensional Engine

PowerOLAP does incorporate a multidimensional database engine as the optimal environment for sophisticated modeling. However, PowerOLAP, by seamlessly accessing relational data, overcomes the shortcomings of predecessor MDB products. The architecture of PowerOLAP's component tools will be fully described in the next section; for now, it is worthwhile to address the points listed previously concerning MDB "drawbacks":

1. In PowerOLAP, data is not "re-created redundantly" from the relational database—rather, it is available for true multidimensional analysis in *real time, as transactions are recorded.*
2. The limit on storage that hampers predecessor MDBs is not relevant to PowerOLAP. PowerOLAP can access only that data which is necessary for immediate modeling; and PowerOLAP has the capability to store models to the relational database to which it is linked.
3. Because of its seamless integration with relational databases, PowerOLAP loses the "proprietary" label that attaches to predecessor MDBs. The data processed in the PowerOLAP engine is as accessible as all data is in an RDBMS. PowerOLAP is effectively the first *open-architecture* multidimensional database in the MDB OLAP marketplace.

Another product difference between PowerOLAP and MDBs should be noted: unlike most other MDBs, PowerOLAP does *not* precalculate all derived values in its multidimensional cubes. Most MDBs follow a precalculation strategy because, in theory, it can provide faster results: if many users typically ask for the same information, the calculated value is already available. The drawback to this strategy is that the most sophisticated models contain many calculated results, so as input values rise, derived values rise exponentially (known as data explosion). Further, when changes to the data are made—quite often, in most analysts' offices—a user must request that the database be calculated again, and this can take valuable time.

The PowerOLAP multidimensional engine calculates values on demand, the optimal approach for a database that receives dynamic updates from a transaction processing system. PowerOLAP maintains a fast response time by storing data in memory and by using incredibly fast algorithms which are tuned to the sparse models characteristic of business analysis scenarios.

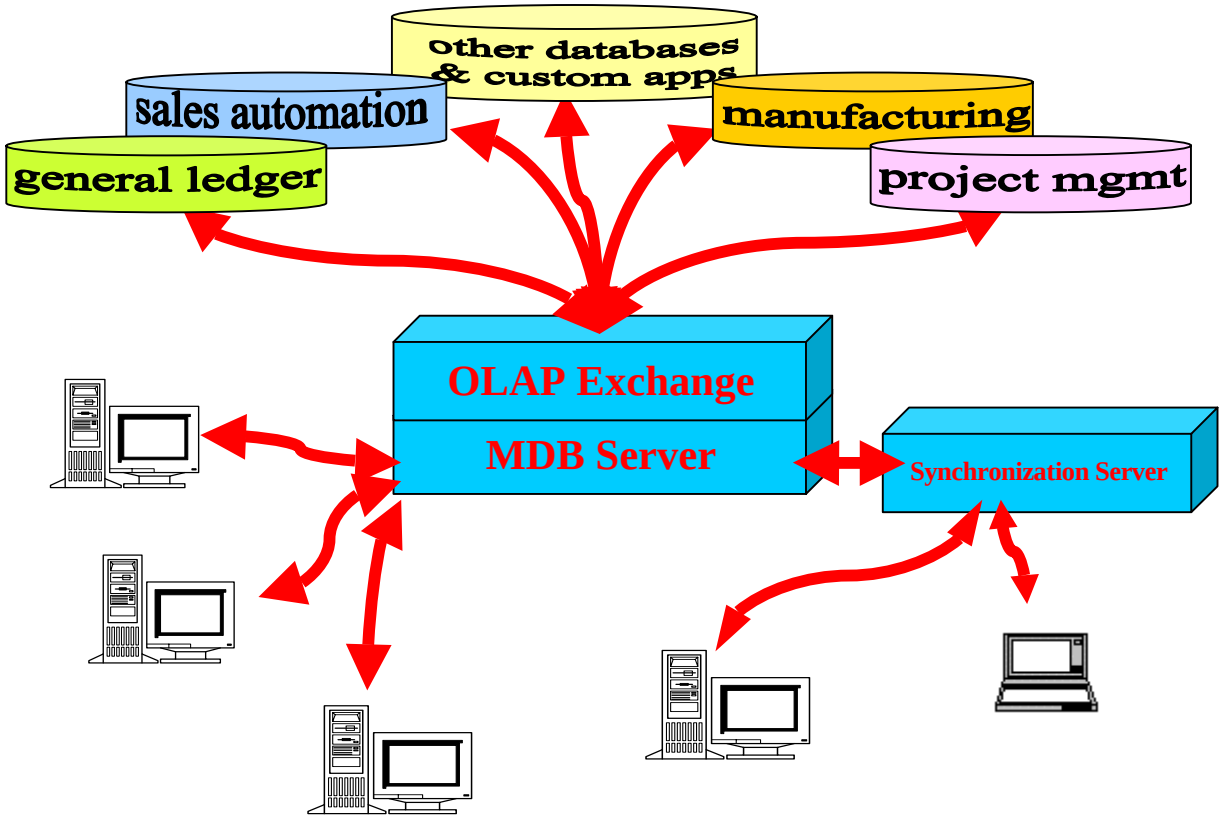
The Power of PowerOLAP™ (1) — Dynamic Bi-Directional Exchange with an RDBMS via *OLAP Exchange*™

OLAP Exchange™ is the PowerOLAP component tool that establishes a dynamic bi-directional data exchange enabling users to "pull" or "push" data between relational databases and multidimensional models.

OLAP Exchange is one of the key component tools that set PowerOLAP apart as a next-generation decision support product. OLAP Exchange provides a seamless "missing link" OLAP technology to establish a dynamic, two-way connection between the previously incompatible worlds of relational and multidimensional databases.

The link from the PowerOLAP client to a relational database(s) is established via ODBC (Open Database Connectivity), a standard database access method. OLAP Exchange then creates necessary tables within the relational database for both relational-to-multidimensional and multidimensional-to-relational exchange of data. Either data exchange can be set up to synchronize periodically, "on demand," or live—provided the relational system supports "triggers". Once the ODBC link is established and the synchronization period selected (in a few simple steps), live and periodic updates occur automatically; "on demand" updates require clicking a single button in a dialog box.

The following diagram shows the architecture of PowerOLAP with OLAP Exchange, and the bi-directional flow of data via OLAP Exchange between PowerOLAP clients and various relational databases.



The benefits of OLAP Exchange are as follows:

- OLAP Exchange directly addresses the need to reach *up-to-the-instant* transactional data located in records databases, and to use that data to budget and forecast, analyze and report, *online*, in *real time*. PowerOLAP represents the ultimate empowerment of end-users, giving them fast, accurate, *current* information from all organizational sources *direct to their desktops.*
- OLAP Exchange means an end to batching, importing or duplicating data between separate database applications; no longer will staff waste time—and cause errors—by manually inputting/re-keying data into various applications, including spreadsheets.

- By allowing incremental updates of data, OLAP Exchange eliminates the time-consuming need to entirely rebuild multidimensional cubes upon each change or update that occurs in the relational source.
- Through the use of OLAP Exchange's Drill-Through feature, users can see the individual records in the source relational database that make up a point in a multidimensional cube.
- MIS groups use a single set of advanced tools, deploying OLAP Exchange with great speed and ease, to get data to end users, rather than utilizing various unsatisfactory methods to update disparate systems or to create and re-create business models.
- PowerOLAP supports an open-architecture environment, since it will integrate any relational data source that is ODBC-compliant via OLAP Exchange.

The Power of PowerOLAP™ (2) — Synchronization of Multidimensional Databases through an RDBMS via *Synchronization Server*™

PowerOLAP's Synchronization Server™ (also shown in the preceding figure) is the component tool that enables the synchronization of multidimensional database models through a relational database, so that users can work "unplugged" and then share and update data from remote locations;

The Synchronization Server, like OLAP Exchange, works via an ODBC connection and adds another powerful tool to establish a dynamic relational/multidimensional connection—in this case to leverage an installed relational system for the purpose of synchronizing independent PowerOLAP clients or server.

Each PowerOLAP client acts as a stand-alone application with the entire multidimensional database loaded into its local memory. A client then writes a change(s) to his or her application and, using the Synchronization Server component tool, sends the change(s) to a shared log table in an enterprise relational database system. Subsequently, other PowerOLAP clients—or Servers—connect to the relational database to read all changes made in the shared table since their last visit.

Several powerful benefits result from this unprecedented functionality.

- There is an organizational benefit: by employing the Synchronization Server tool companies leverage their investment in already-installed relational systems.
- By working through a shared table, clients can work "unplugged"—at home or on the road—and then connect to the relational system to upload changes they make, or to download changes others have made for full synchronization of models throughout the organization.
- Because Synchronization Server stores entire PowerOLAP models—and ongoing incremental changes—in a relational database, it can work in concert with the MDB Server as an online relational backup to multidimensional data. Thus, should the MDB Server be compromised, Synchronization Server would be available to replicate PowerOLAP databases onto local machines or even on another MDB Server. Further, Synchronization Server can act as a "load balance" between two MDB Servers, sharing changes between them in the same way it synchronizes data between individual clients.

The Spreadsheet as a Front End—the Dynamic PowerOLAP/Excel Connection

PowerOLAP employs Excel as a front end, one of its most "user-friendly" features, but also tremendously practical for enterprise tasks like performing roll-ups of departmental budgets. But locating PowerOLAP on an MDB Server, multiple users can view report pages or contribute figures via the desktop tool they use most often and know best—Excel.

With the click of a button, a parametric "slice" view of a multidimensional cube creates a spreadsheet—a fully functional spreadsheet, able to create graphs and employ formulas (both Excel- and PowerOLAP-based formulas). Then, from within the spreadsheet, it is possible to enter figures, change parameters of the view or perform higher-level analytic functions. Most important, the spreadsheet remains *dynamically connected*, via PowerOLAP, to underlying data stores. Thus the spreadsheet will update automatically as the underlying figures or structure changes, providing Excel—and end-users—with an unprecedented degree of empowerment, directly, at the desktop.

Summary

PowerOLAP enables organizations to create dynamic solutions quickly—in hours rather than days—between relational and multidimensional databases. With OLAP Exchange, users have *true online* access to transactional data—*as that data is recorded in real time*—within a multidimensional format. PowerOLAP employs other components as well to provide out-of-the-box solutions for true, online, real-time analytical/ reporting/ budgeting capabilities. In all, PowerOLAP represents the "next-generation" advance of OLAP technology.

PowerOLAP leverages investments and knowledge in existing database systems, providing unprecedented flexibility and speed for the back-end integration of transaction and analytical systems. By employing Excel as a front end, PowerOLAP connects users throughout an organization with underlying data sources via the tool they know best, the familiar spreadsheet, direct to their desktops—another example of how PowerOLAP leverages company investments and knowledge.

With real-time transaction data at their fingertips, analysts are far more productive: the business models they create contain up-to-the-instant information; they no longer waste time keying in figures or trying to reconcile data from various sources, including individual spreadsheets.

PowerOLAP is one single solution—*finally*—that closes the gap between relational and multidimensional worlds, making advanced technology available to everyday analysts for powerful, real business benefits.