INVESTMENT RESEARCH AND PORTFOLIO MANAGEMENT ANALYTICS USING ORACLE OLAP

Chris Claterbos, Vlamis Software Solutions, Inc.
claterbos@vlamis.com

Peeyush Shukla, CFA, FRM, Mesirow Advanced Strategies, Inc.
pshukla@mesirowfinancial.com

INTRODUCTION

Mesirow Advanced Strategies, Inc. (MAS) is one of the largest hedge fund of fund managers in the world. Today, MAS manages more than $16 billion in client assets.

MAS implemented a suite of OLAP applications to help facilitate its investment research and portfolio management process. These applications have helped MAS in successfully achieving the following goals:

- Ability to analyze the universe of equities, corporate bonds, and hedge fund managers; and various MAS proprietary portfolio measures along several key dimensions of interest.
- Ability to integrate relational data and multidimensional cubes in a single platform that is provided by Oracle Enterprise edition with the OLAP option.
- Leverage the integrated power of Oracle's relational and multidimensional engine to develop an efficient, flexible, and scalable solution that aggregates complex hierarchies quickly, and ensures optimal query performance on a large data set that is concurrently accessed by many users in unpredictable ways.

This presentation provides a technical overview of the solution that was developed on the Oracle OLAP platform, using the Oracle business intelligence toolkit.

OVERVIEW

Companies need to analyze their information in ways that enable decision makers to respond quickly to changes in the business climate. While a standard transactional query would ask, "How many bolts were sold last month?"; an analytical query would ask, "How do sales in the Midwest for the last 3 months compare with the forecast? How does that compare to a year ago?" Such analytical queries require an online analytical processing (OLAP) solution such as that provided by Oracle 10g:

- The Oracle relational database is one of the most efficient and safe ways to store information. A Data Warehouse is structured to store data in a form that facilitates efficient reporting and analysis.
- The OLAP Services provide a Java OLAP API and an analytical engine to build analytical applications that support complex statistical, mathematical, and financial calculations along with predictive analytical functions such as forecasting, modeling, consolidations, allocations, and scenario management. A java-based architecture allows OLAP Services to support deployment of analytical applications to large, geographically distributed user communities over the Internet.
- The Oracle BI Beans component complements OLAP Services by providing pre-built Oracle JDeveloper or other Java development environments to build analytical applications that can be deployed as either Java or HTML ("thin") clients.

The integrated OLAP solution has enabled MAS to achieve its goals by building a series of OLAP applications in a phased manner. These applications are used by over 30 users currently, and the plan is to double the number of users within the next six months. This has also helped MAS in retiring some Matlab and Excel-based analytical applications, by allowing drill-down and drill-through capabilities to a large amount of detailed data, compared to what was originally possible.

TECHNICAL SOLUTION

The OLAP application suite at MAS has been developed in a series of phases. The initial application was comprised of 9 dimensions, and 4 cubes that contain over 12 measures. The application suite has now grown to four applications, comprised of 26 conformed dimensions, and 9 cubes that contain over 45 measures. Many measures contain complex calculations and/or aggregation methods that are most efficiently implemented using the MOLAP engine. Cubes are updated at various intervals –daily, weekly, and monthly; and even with several million records, it only takes anywhere from 30 minutes to a few hours to load and aggregate this data. Over time, query response times have improved to a point where just about any query can return a result in a few seconds.

The architectural components of this application include Oracle Database10g with OLAP 10g, Oracle Warehouse Builder (OWB) 10g, Oracle Application Server, Analytical Workspace Manager (AWM), JDeveloper and BI Beans. Oracle Warehouse Builder and Oracle Analytic Workspace Manager were used to build and manage the processes to load data from various internal and external data sources, and ultimately into the OLAP cubes. Security is implemented at the measure and cell level for each database user. The front-end was developed using Oracle JDeveloper, and Oracle BI Beans provides the visualization components for the web based application.

ARCHITECTURE

The application design is based on a multi-tier approach. The components of the architecture are shown in Figure 1.

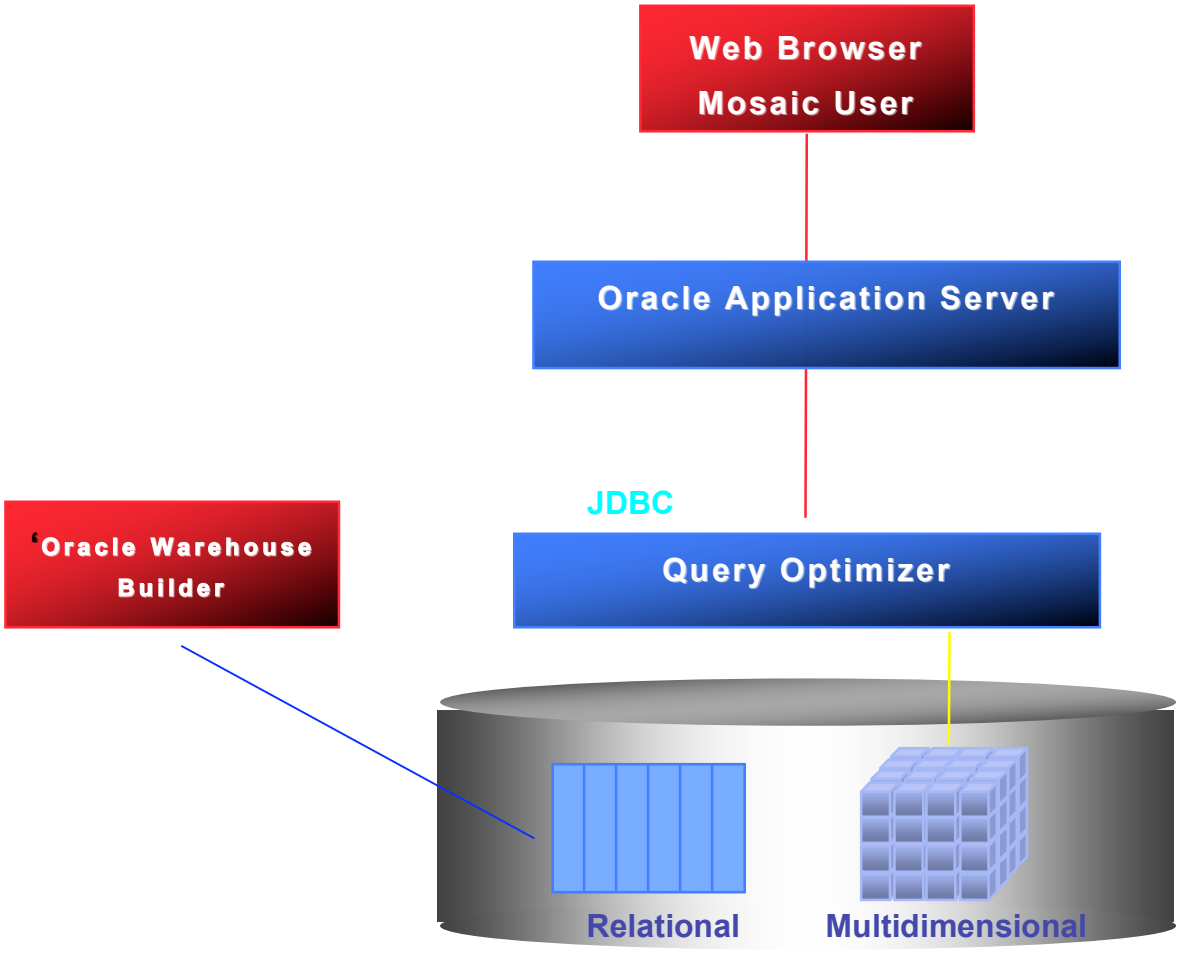

Figure 1. MAS OLAP Architecture

BACK-END ANALYTICS

The relational star-schema based data warehouse was designed using the Erwin data modeling tool, and populated using Oracle Warehouse Builder (OWB). At the time of development, the version of OWB selected to populate the warehouse did not fully support construction of OLAP-based models. As a result, it was decided to use AWM to model and construct the OLAP metadata and data objects. AWM allowed for creating flexible cubes and dimensions with complex aggregation rules and calculation formulas. Some of the calculations and aggregations still required custom coding to achieve the necessary results. This coding was done using the OLAP DML language supported in the Oracle OLAP option. The language allowed for creation of programs that would perform complex multidimensional calculations and summarizations.

DESIGN AND IMPLEMENTATION CONSIDERATIONS

There are several issues that need to be considered when building OLAP and BI Beans applications. Once the back-end data models were developed, it became necessary to construct a front-end framework that could be used to support not just the first application but a series of applications that would build upon the initial application and provide the end-user with a consistent interface into the data. Early on in the project, it became evident that the end-user query requirements would require some custom user controls, so that data could be quickly selected and displayed. MAS decided to use the Thin Client architecture for the front-end application framework because of its simplicity, fast performance, and zero client footprint requirements. Figure 2 shows a sample screenshot of one of the OLAP applications.

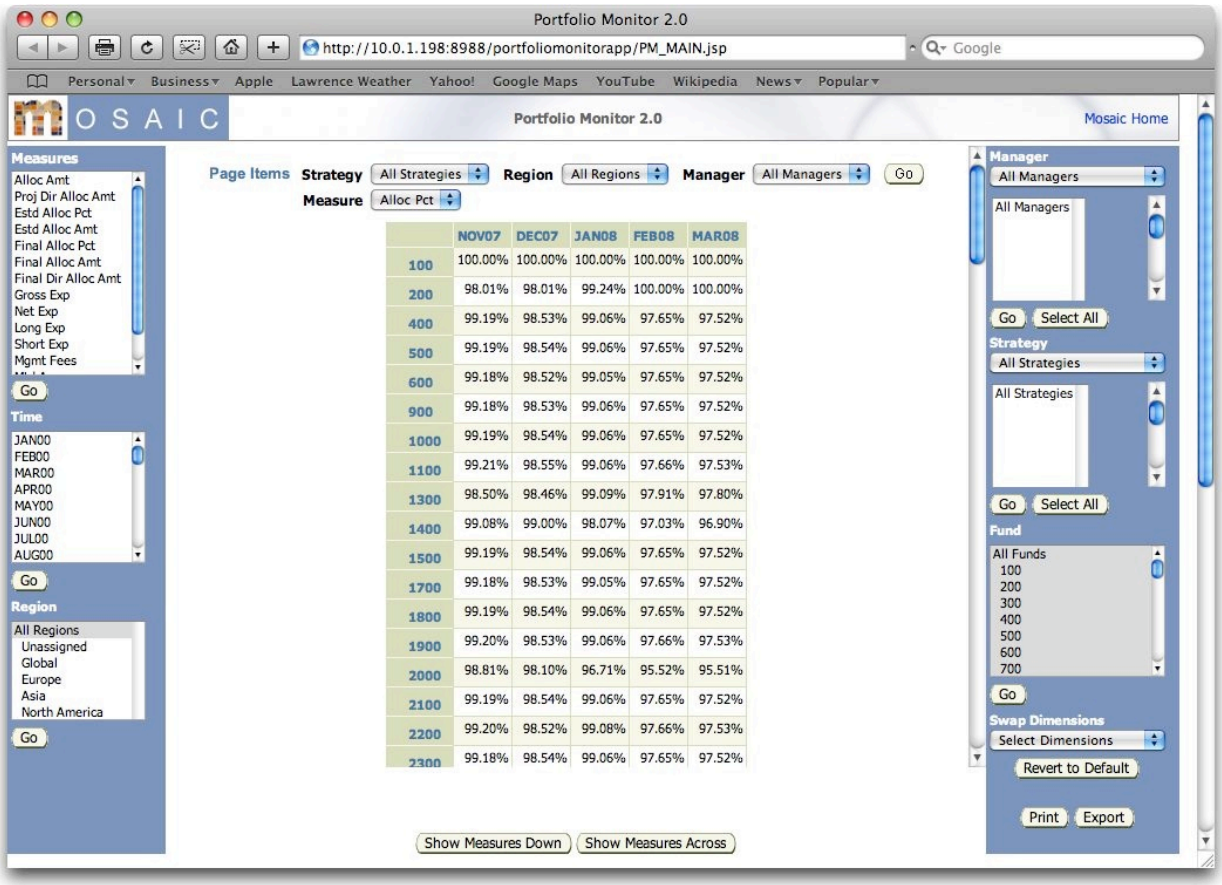

Figure 2. MAS OLAP Application screenshot

GOING BEYOND THE WIZARDS

While the wizards provided with BI Beans allow for creation of a basic working application that provides an excellent starting point, the requirements for MAS went beyond the basic functionality provided by Oracle BI Tags. This required the creation of several custom classes that extended the basic BI Tags to provide the desired functionality. This included a Date Range tool and a Dimension level tool for selecting dimension members by level. It was also necessary to provide a tighter synchronization of the front-end data status with the back-end as business users limit the data to be viewed. A java framework was also created to allow for addition of new applications to the suite in an expedited manner, greatly reducing the costs and times to bring a new application on-line; while also making enhancements to existing applications easier. A common library approach made enhancements easier by simply recompiling existing applications to take advantage of the new features.

CONCLUSION

The powerful suite of OLAP applications at MAS has provided an extremely flexible analytical solution to help analyze over forty five investment research and portfolio management measures along the key conformed dimensions of business. The complexity and sophistication of query requirements could only be accomplished via a combination of Oracle's relational capabilities to warehouse a large amount of data, combined with the power of its multidimensional engine to enable unlimited ways of viewing this data.