Comparing and Contrasting Online Analytical Processing (OLAP) and Online Transactional Processing (OLTP) Architectures
Introduction

One of the most important assets that an organization possesses is the data that it collects. In order to remain competitive in today’s extremely dynamic business environment it is often critical to have a panoramic view of an organization’s health through data analysis. By performing data analysis, managers and other business decision makers can gain important insight that can facilitate developing business goals, assist in understanding deficiencies and provide a catalyst in evaluating business performance. In order to achieve the goal of obtaining a comprehensive view of the data within an organization, it is paramount to have in place a technology infrastructure that can support data analysis. Many organizations have implemented an On-Line Analytical Processing (OLAP) system to help fulfill this business analytic need. Unlike a traditional relational database model or On-Line Transactional Processing (OLTP) system, an OLAP system is optimized to provide data to end users in a meaningful format through a Decision Support System (DSS) Application. In order to offer an environment where a DSS can effectively provide analytical functionality an OLAP architecture must differ from a typical transactional based system. Some of these differences consist of data dimensionality levels, variations within query and transaction procedures, along with diversity within the database size, data volatility, granularity, and physical database design. Due to these differences it is imperative to ensure that these two architectures remain autonomous to ensure performance and functionality are maintained at acceptable levels.
OLTP vs. OLAP

One of the most prominent differences between OLTP and OLAP systems is in regards to dimensionality (Rob & Coronel, 2002). Business analysts are interested in viewing data using a variety of perspectives. For example, a decision maker may be interested in understanding the relationship between sales representatives, the products sold and the regions in which the products were purchased within a given time frame. In this circumstance, the data would be viewed by the analyst across several dimensions. E. F. Codd described the business users desire to merge data in endless combinations “slicing and dicing” data (Codd & Salley 1993). OLAP systems are designed to provide a multidimensional view of the data as described above. This is accomplished by housing the data in a schema with a relatively few number of tables where data can easily be grouped or joined. Whereas a transactional system typically is concerned with individual records and usually consists of a higher number of tables that often times have complex unintuitive relationships. A transactional system will generally utilize an application oriented or entity relational database model, while an OLAP system will oftentimes employ a subject oriented model that stores data in a simple star or snowflake schema.

Another area in which OLAP and OLTP systems differ from on another is how they are designed to handle transactions. An OLTP system has been configured to handle a high number of updates or insert statements. For example, an online retailer may have a database that collects data on customer orders that are placed for products listed on a website. This database may be updated with new orders thousands of times a day, whereas updates within an analytical system occur very infrequently. A typical OLAP
system might be updated once a day in order to obtain the most recent transactions from the previous day. Therefore, transaction speed would not be a critical concern within an OLAP designed system, but would be extremely important when working with a transactional based system. It may be very common for a user of an OLAP system to submit an extremely complex query where the database management system (DBMS) is forced to access millions of rows in order to return a set of results. Hence, querying speed is of utmost importance within an analytical system. Conversely, querying performance is much less important when working with an OLTP system.

In relation to multidimensionality, OLTP and OLAP systems also differ in the degree to which they are normalized. Generally speaking, transactional systems are prone to problems with data anomalies that are associated with a high degree of insert and update statements (Rob & Coronel, 2002). Hence, a high degree of normalization is deemed important within these systems. However, in the case of an OLAP architecture, a large amount of inserts and updates are not typical. It is the purpose of an analytical processing architecture to facilitate querying, therefore, a denormalized database model is preferable in order to make extracting meaningful data fast and efficient.

OLTP and OLAP systems also differ from one another in regards to volatility (Elmasri & Navathe, 2004). As discussed earlier, analytical systems are generally updated less often than a transactional system, hence, the analytical system does not contain real time data. Depending on the needs of the business user, the data contained within an OLAP system may be weeks if not months old. This observation helps accentuate the fact that analytical processing systems are not concerned with atomic transactions, but more concerned with a macroscopic, historical perspective of the data.
For example, a business analyst probably would not be as interested in knowing that a given sales representative sold a specific product on a specific date. However, understanding the sales ranking for a specific product over the last two years may be more helpful in making decisions involving the company’s strategic goals.

Data summarization and granularity is another area where analytical and transactional architectures diverge. OLTP systems collect data at a very granular level, taking into account every last transaction. However, as discussed earlier, OLAP systems are more concerned with a wider perspective, hence, oftentimes data is summarized in analytical systems. For example, a transactional system may write a row of data each time a product is sold. However, an analytical system may only attempt to store aggregate information regarding product sales, such as the number of sales a day for a given product (Rob & Coronel, 2002). Again this type of information would better facilitate a decision support system. Online analytical processing systems also can be designed in a manner to facilitate roll up and drill down functionality. A user may be required to view data at a particular level, at which point, it may be important to analyze the data at either a more or less granular level. For example, a marketing manager may need to view product sales by month, however, it also may be pertinent to have the ability to roll up and analyze product sales by year, or perhaps drill down and view product sales by week (Rob & Coronel, 2002).

Database size can also vary when comparing OLAP and OLTP systems. Generally speaking OLTP systems are smaller when compared with analytical systems. First, to optimize a transactional system, many database administrators will purge or archive data periodically resulting in smaller space consumption. However, since
analytical systems provide a more comprehensive historical record, they are rarely purged. Second OLAP systems are not normalized structures. This results in data duplication, which will invariably result in an increased size. Lastly, data in analytical systems can exist in various alternate forms. For example, it may be necessary to store call handle time within a data warehouse for a call center. In addition it may also be important to store information regarding call handle time range that is calculated using the initial call handle time fact. Call handle time range and call handle time are calculated using the same data set, however one is stored in an aggregated capacity in order to provide business users with a more useful set of data. This practice is another factor that results in a larger database size when compared to a transactional system.

It is also appropriate to note that the user profile also varies between transactional an analytical architectures. Transactional based systems have many users; these users usually consist of a generalized customer base (through the use of an application), a clerk or other informational technology professional. Conversely, the typical user of an OLAP architecture is a business analyst, executive manager or some type of company decision maker. The users of the analytical based system are relatively few when compared with the transactional user base.

In addition to the differences described above, another characteristic of an OLAP architecture is that it must be easy to understand (Kimball & Ross). Since users depend on this system to identify and comprehend correlations between business entities it is imperative that the system and the relationships found within the system are easy to grasp conceptually. If the system is conceptually difficult to identify with, unfortunately the fundamental purpose of presenting useful data to the end user is lost.
Importance of Autonomy

Many companies are extremely interested in obtaining the benefits that a decision support system offers. However, it is tempting to consider short cuts when taking into account that building an analytical system on top of existing transactional system can require a great deal of additional resources to manage and implement. One short cut that some companies consider is attempting to get their decision support system to run on an existing OLTP architecture. This can result in many problems from both a functionality and performance standpoint (Navathe, 1999). Operational databases are specifically tuned, and indexed to run off a known workload and have potentially been optimized to work in tandem with a front end application. Adding a DSS to this system could result in an imbalance resulting in performance degradation. In addition, since OLTP systems are highly normalized, complex queries can be extremely taxing on the system due to the high number of joins required within a relational model. This can also result in performance problems. Lastly, many decision support systems are unable to easily follow complex joins typically found in the relational model. This provides an environment in which the DSS is running inefficiently or inaccurately. Potential functionality problems that exist when attempting to use a transactional architecture for analytic purposes include both missing data and unconsolidated data. If data has been purged from the system, the DSS could portray inaccurate or incomplete results. Also, since DSS systems often need consolidated or summarized data to function properly, a transactional system could again result in a non-functional system or a system that offers erroneous results.
Summary

An OLAP architecture offers an environment in which business users can perform historical, multidimensional time-based analyses in order to help facilitate good decision making within an organization. (DataBecon, 2004) There are several differences to consider when comparing OLTP and OLAP systems. Some of these differences include the way the physical database is constructed, the methods in which historical data are stored and the type of users involved with each system. Additionally, it is critical to ensure that transactional and analytical systems are separated from one another so that performance and functionality are maintained. Overall, OLAP and OLTP systems each serve a distinct purpose and when used correctly, can be an extremely powerful tool in establishing or maintaining a successful business.
Literature Cited


