P: ISSN NO.: 2321-290X RNI : UPBIL/2013/55327 **SHRINKHALA :** VOL-II * ISSUE-I*September-2014
E: ISSN NO.: 2349 – 980X

Bridging the Data Warehouse Gap in the Real Time Interactive Information Organization

Abstract

This article is a position paper on the Business Intelligence (BI) technology has helped many organizations make better decisions and improve performance. Unfortunately, implementing BI does not automatically lead to better results. Too often, BI systems deliver only a small fraction of their potential value. Experience has shown that the way in which BI technology is implemented has a huge impact on results. One of the most important keys to success is the use of a data warehouse as the foundation for a BI solution. The way in which the data is brought into the warehouse is extremely important. The most effective approach keeps the data in the warehouse closely synchronized with data from source databases – something often referred to as real-time updating.

Even though the advantages of real-time data warehousing have been known for some time surprisingly few working BI solutions employ it today. A key reason for limited use of this very valuable approach is a lack of understanding of what it is and what is involved in accomplishing it. Fortunately, improving technology and lessons from the experience of those that have successfully made it work have combined to make real-time data warehousing practical and affordable.

This paper will explain real-time data warehousing in simple terms and will provide practical guidance on how to make it work. The advice offered here is not theoretical. It has been successfully used for many years in a wide variety of complex organizations

Keywords : BI, SDLC, CDC, ETL

Raj Kumar
Ph. D (IT),
Research Scholar,
Department of Mathematics,
BRA Bihar University,
Muzaffarpur, Bihar,

Ajay Kumar Singh
Department of Mathematics,
BRA Bihar University,
Muzaffarpur, Bihar,

Introduction

A Data warehouse is a composite and collaborated data model that captures the entire data of an organization. It brings together data from heterogeneous sources into one single destination. It is not just bringing together. Data is Extracted, Transformed and Loaded (ETL) into the Data warehouse. This processing of the data is usually done in what is known as a 'staging area'. The data need not be normalized, as it will be mainly for read-only purposes or basically querying and analytical purposes. Data warehouses are mainly OLAP systems and they do not operate on OLTP data. Enterprise data comprises of multiple data residing in multiple systems. Data will be duplicated at many places; however it facilitates easy day-to-day OLAP operations. Inmon, Father of Data warehousing defines a Data warehouse as subject oriented, integrated, non-volatile & time variant collection of data in support of Management Decision.

Mergers and Acquisitions are very common in today's business arena. Every day, there are talks going on between retail giants, players in banking and insurance domains. Every quarter there are some fruitful mergers or acquisitions. What does this mean to the IT departments? Simple - Extraction, Transformation and Loading of voluminous data! There is a huge amount of Organizational data movement. Testing ensures data integrity during data movement and the data loading.

Today Senior Managements rely on Decision Support Systems for policy-making in the organization. Hence, the DSS (and the underlying data model) should have the ability to monitor historical trends, patterns and provide suggestions/conclusions. Monitoring near real-time operational data also provides the organization with the much coveted completion-advantage. It's a basic rule of thumb that the Defect detection should be as early as possible in an SDLC. This rule has utmost significance when it comes to DSS and Management Information Systems. Any defect in the data model, the warehouse implementation, the Extraction or the processing is capable of translating into disastrous decisions by the Organization.

Financial Institutions, Health Care institutions etc are made to

comply with stringent IT policies in reference to the customer data and day-to-day transactions. Implementation of such regulatory applications may mandate ensuring of the compliance on the historical data as well. The most effective BI solutions are based on data warehouses. The only other option – using BI tools to obtain data directly from source applications – has been shown countless times to simply not work acceptably. A data warehouse maintains a copy of data from sources that usually include the ERP software that runs business operations. A data warehouse runs on dedicated servers and organizes and formats the data in a way that facilitates analysis and reporting.

In a perfect world the data in the warehouse would instantly reflect changes in source databases. In practice some delay is inevitable. Experience has shown that a delay of up to a few minutes in synchronization of a warehouse with its sources is almost always acceptable. Longer delays, however, can create problems. This paper will refer to any data warehouse that maintains a short and acceptable delay in synchronization as a real-time data warehouse.

Techniques for achieving real-time synchronization have been available for some time. Unfortunately, they have not yet achieved widespread use. Applying them has become practical and affordable and thus should be the first option explored by any organization starting down the BI path. Retrofitting existing BI solutions with a real-time data warehouse is also an option worthy of examination.

Most data warehouses in use today are not real-time. Instead they typically synchronize with data sources once a day, usually late at night. Data warehouses updated this way often use the brute force approach of re-copying everything in the source databases. This daily refresh approach takes a long time and can create a serious scheduling problem since the source servers normally need to stop all other activity while this is taking place. The only other option is to provide the excess processing power needed to accommodate the refresh.

The technical term for the process used to achieve real- time synchronization is Change Data Capture or CDC. A number of ways to achieve CDC have been tried over the years. Only one approach has stood the test of time and has proven to be acceptable. It is based on obtaining data from log files created by the source computer's database management system. Other approaches that have been tried and found to be inadequate will be examined later in the more technical section of this paper.

The best practice for BI is thus to use a data warehouse that employs log-based CDC for updates. Those evaluating data warehouse options therefore need to verify that any claims of a CDC capability are log-based and do not employ one of the inferior alternatives.

Log-based CDC is important because it is the only updating technique that can insure that all changes to the source databases will be reflected in the data warehouse. Without this assurance the value of the data warehouse and the output it generates can be greatly diminished.

Those using a data warehouse need to have confidence that its contents are complete and accurate. Many BI projects fail because users lose confidence in the data warehouse and stop using it.

A growing number of BI systems produce output that is vital to the ongoing success of the organization.

Financial results are often reported to the public and to regulators based on information in data warehouses.

The updating technique for these warehouses must be unassailable, auditable and in compliance with Sarbanes- Oxley regulations. Log-based CDC is by far the best way to achieve this. The more detailed examination of how this approach works is presented later in the paper and will make it clear why this is the case.

Most data warehouses are built using a type of software tool named for what it does: Extract Transform and Load or ETL. Popular tools in this category include Informatica PowerCenter, IBM InfoSphere DataStage, SAP Business Objects Data Services, and Oracle Data Integrator. Other options are available from less well known vendors. All perform the basic functions of obtaining data from a source server (extraction), reformatting it (transformation) and updating of the resulting data warehouse (loading).

Not all of these tools can perform log-based CDC on their own. ETL tools that do not support this function need to be combined with another software tool that adds this capability. The major vendors have tended to acquire software firms that offer a log reading capability and then either integrate it into their ETL tools or offer it as an optional complementary product. Those evaluating data warehouse offerings cannot therefore assume that vendors whose product line includes a log-based CDC capability will automatically include it in a proposed solution.

One of the reasons that log-based CDC is not more widely used is that the use of this technique has implications on the way the data is organized and stored in the warehouse. That structure is referred to as the Data Model of the warehouse. If the Data Model does not follow certain design principles then the performance of the resulting data warehouse can be disappointing.

Those about to undertake a BI project need to fully understand the impact that real-time updating will have on the Data Model design. If a warehouse is first implemented using daily batch updates then a significant redesign might later be necessary in order to make log-based CDC workable.

The advantages of real-time updating

An obvious advantage of real-time synchronization is that there is no need to reconcile differences between source applications and the warehouse. A data warehouse that does not appear to always be up to date tends to lose the confidence of the user community. Lost user confidence can lead to the failure of a BI effort.

This is because the return on investment in BI is dependent on how many different things it is used for and their collective impact. Any loss of confidence in the quality or timeliness of the data on which output is based reduces the use of BI and thus its impact.

A well designed log-based warehouse will be fully auditable. This makes it possible to create reports and other output that is provided to outsiders such as investors or regulators. The usage and thus value of a BI environment based on auditable data tends to be much

higher than one whose data cannot be fully trusted.

Another major potential benefit of BI is the elimination of debates over whose data is accurate. This benefit is hard to realize unless the foundation of the BI solution is a real-time data warehouse.

BI systems can also contribute greatly to operational efficiency if they include the ability to initiate action-oriented alerts. Traditional applications often cannot instantly analyze the impact of unusual transactions such as a large rush order, a delivery delay, or a major purchase price variance. BI systems can be programmed to identify the need for action in many of these situations and to alert the appropriate personnel, often through dashboard displays. Without the support of a real-time data warehouse, it is not possible to create this type of action-oriented alert.

Log-based CDC also represents the lowest impact method to update a data warehouse. Every other approach puts a greater burden on the servers supporting the source databases. Performance is thus improved and unnecessary costs are eliminated.

Summary

BI solutions that employ a real-time data warehouse with log-based updating will tend to deliver far more satisfactory results over time than those without it. Other approaches will not perform as well, may not be reliable enough to be auditable, and will create confusion at times due to timing differences.

Proven techniques have been established to make real- time data warehousing practical and affordable. A small but growing number of pre-built log-based real-time data warehouses are available for purchase. If one is available for the sources you use then it should be seriously considered.

When a pre-built solution is not available it is usually necessary to either custom-build a warehouse or engages others to create it. In either case it is important to involve experts in log-based CDC. It is also important to insure that the Data Model on which your data warehouse is created anticipates real-time updating.

Over time the highly effective use of BI is likely to become a competitive necessity for every complex organization. In anticipation of that, it is appropriate to set a goal of working towards real-time data warehousing. Achieving that goal may not be as difficult as it now appears. A more detailed understanding of exactly what is involved may be useful

Change Data Capture

Every data base management system (DBMS) includes a mechanism to save data for backup and recovery and to support commitment control. All use a similar approach based on the creation of log files. Each time a change occurs in a database under its control, the DBMS insures that a complete record of the change is written to a log file. This file can be stored on devices that are separate from those used to store the source data so that if one of those devices or its server fails, the log remains available.

The commitment control process allows databases to insure that data is never lost due to failure during the process of updating or otherwise changing data on the source system. The log contains records of changes that have been confirmed to be complete by the commitment control process.

Log files contain a complete record of all changes that were made along with a date and time stamp indicating exactly when each change occurred. These log files make it possible to repair the source database in the event of failure. They are used in the process of recovering from a device failure (or software error) by restoring the data on a device to a copy made at a particular time and then re-processing transactions from the log files until the device is restored to the point it was when the failure occurred.

The key point is that database log files are designed to reflect every possible change in the database. Creative programmers, hackers, hardware crashes or even cruel fate cannot get around making them the best possible source of data on which to base a CDC strategy.

A mechanism similar to the one used for device recovery can therefore be used to create a data warehouse that is synchronized with its data sources without fear that changes will be missed.

How it works

An examination of the way log-based CDC works should make it clear why this approach is vastly superior to any alternative. The starting point is the ETL tool that is used to organize and manage the entire process of building data warehouses. Some ETL tools do not have the capability to perform the extract function when the source data comes from log files instead of the source database itself. Such tools need to be combined with another tool that does offer this capability if log-based CDC is going to be implemented.

It should be noted that each DBMS has its own unique way of creating log files. A driver or software interface to each DBMS from which source data needs to be obtained will thus be needed. Commonly used DBMS's include Microsoft SQL Server, Oracle 11g, MySQL and IBM DB2.

Design Considerations

Transformation and loading are complex procedures that take time and computer capacity. There can be occasions where an unusually heavy volume of change transactions can cause the ETL process to fall behind the log file. In these cases update transactions are not lost, just delayed.

Designers of log-based CDC data warehouses need to determine how much server capacity to put in place to handle such intermittent bursts of change traffic in order to limit the duration and frequency of delays.

Experience has shown that most data warehouse users will not notice or be concerned about synchronization delays of a minute or less. Most will tolerate longer delays if they occur infrequently. Real-time data warehouses should be designed to continuously monitor their own level of synchronization and alert administrators and users if pre-set service levels are not met.

A well-designed log-based CDC system will have no noticeable impact on performance of the source servers from which data is being extracted. This is one of the most important benefits of this approach to CDC. In most cases it is very important to avoid slowing down the process of updating source servers since the transactions they

handle are often vital to the ongoing operation of the entity. Log-based CDC is by far the best way to minimize impact on source transaction servers.

Method of CDC

Approaches other than log reading that have been tried, and found to be wanting include:

1. Database triggers – this DBMS feature invokes a prewritten routine each time a specific set of conditions are met, such as the addition or updating of a record in the database. In concept, the code that the trigger launches can be written to write a record of the transaction to a database table and the ETL tool can then poll those tables on a periodic basis. In practice, this approach is far from foolproof. For example triggers can regularly be deleted/disabled then re-added/re-enabled in the normal course of a business operation. Triggers also place a relatively high overhead burden on the source database server
2. Message Queues – a number of middleware products such as IBM MQ Series also appear to have the capacity to capture application, not database, changes and report them to an ETL tool. An obvious disadvantage is the cost of the license for these products. More importantly, message queues are not a completely reliable source of change information since they only know about data changes that are sent to them by the applications and not batch routines or manual updates to the database.
3. Date and time stamps – many ERP applications and other data sources maintain data fields within each record that indicate when it was last changed. An approach to CDC that has been tried a surprising number of times reads through the data records and looks for recent changes. The fatal flaw in this approach is that it relies on the programs that change data to unfailingly update this field. In addition, this approach can lose track of deletions since the entire record, including the time stamps, are gone.

Sadly, some of the major BI vendors claim to update data warehouses in real-time but do so using an inferior approach to log-based CDC. Buyers need to look under the covers and verify that properly designed log-based CDC forms the foundation of any data warehouse before it is put in the hands of users.

Conclusion

When a real-time data warehouse is being created the first step is a one-time process to populate the initial version of the warehouse. The ETL tool finds and extracts every data item from the source databases, processes all the necessary transformations and loads everything into the data warehouse. This start-up process can take hours depending on the size of the files involved and the power of the servers. It is important to note that this time consuming process only needs to happen once in a CDC-fed data warehouse.

With the log-reading drivers in place the ETL/CDC tool is turned on. The log readers' poll the log files and use them to identify all changes that have occurred. These changes are sent to the parts of the ETL tool that handle the transform and load functions. When the ETL tool is ready to handle the next wave of update transactions another polling cycle is initiated. The polling cycle typically takes place every few seconds and can be varied as appropriate.

The log reading logic thus handles the extraction phase of the ETL process. It feeds change transactions to the transformation function. This is the step where source data is made more understandable and usable by BI analysis tools. Often this involves changing the names of data fields to make them both understandable and consistent.

The final function of loading the data warehouse can then take place. The transformed data is normally organized in different ways than it was in source databases. Combinations (de-normalization) of tables are frequently performed to make access faster and easier.

References

1. [ASII91] American Supplier Institute, Inc.. Taguchi Methods: Introduction to Quality Engineering.Allen Park, Mich. 1991.
2. [BaTa89] D.P. Ballou and K.G. Tayi. Methodology for allocating resources for data quality enhancement. Communications of the ACM (CACM), vol. 32, no. 3, 1989.
3. [BBBB95] D. H. Besterfield, C. Besterfield-Michna, G. Besterfield and M. Besterfield-Sacre. Total Quality Management. Prentice Hall, 1995
4. [Boed87] R.F. Boedecker. Eleven Conditions for Excellence. The IBM Total Quality Improvement [OiBa92] M. Oivo, V. Basili. Representing software engineering models: the TAME goal-oriented approach. IEEE Transactions on Software Engineering, 18(10), 886-898, (1992).Process. Quincy, Mass. American Institute of Management, 1987.
5. [Wang98] R.Y. Wang. A product perspective on total data quality management. Communications of the ACM (CACM), vol. 41, no. 2, February 1998.
6. www.wikipedia