ETL Scheduling in Real-Time Data Warehousing

REVATHY SREEKUMAR[*]
PG Scholar
Sri Ramakrishna Engineering College
Coimbatore , India
revathysreekumar@gmail.com

SARAVANA BALAJI . B
Assistant Professor
Sri Ramakrishna Engineering College
Coimbatore , India
saravanabalaji.b@gmail.com

Abstract— As the pace of today's business grows and with business getting more aligned with good customer relationships, decisions that are made based on the most recent data are required. These decisions that are based on the most currently available data improve customer expectations and hence customer relationships, increase revenue and maximize operational efficiencies. This technology that aids in taking business decisions on the fly is known as real-time business intelligence. ETL being the core of the Data Warehousing process faces numerous challenges while being implemented in real-time. One of the challenges is to how to order the data being pushed into the ETL process from various sources simultaneously without incurring data loss. This paper looks into how some of the scheduling algorithms available can be modified for use in the ETL process.

Keywords-Real-Time ETL; Real-Time Data Warehousing; ETL Scheduling Algorithms; Real-Time ETL Scheduling

I. INTRODUCTION

Decision making is the lifeline of any business. Whether one wants to arrive at some marketing decisions or fine-tune new product launch strategy, decision making through data analysis is the key to all the problems. Reliable decisions can be obtained only from the analysis of data that is well organized. Merely analyzing data isn't sufficient from the point of view of making a decision. How to interpret from the analyzed data is also important. Thus, data analysis in this notion should not behave as a decision making system but rather a decision support system.

Data within an organization will be kept either in the transactional systems or it will reside within a Data Warehouse [1]. Information within the Data Warehouse is prominently used for business analysis. Data warehousing is recognized as a key technology for the exploitation of these massive amounts of data present in the transactional systems.

The Data Warehouse provides the end users with numerous benefits. Foremost among these advantages is that the Data Warehouse allows the vast amount of data available in various disparate end systems to be accessible at a single location. It allows the data to be organized in a manner that is required by those who actually use the Data Warehouse. Various techniques are employed to ensure consistency of data and to eliminate unwanted redundancies. Access to the data can also be controlled. And the most lucrative of the benefits is that it aids in decision making.

An organization consists of a large number of source systems like files or databases where operational or transactional data are being stored. The Data Warehouse will be updated on a timely basis. This Data Warehouse refresh period can range from once a week to once a day. This refresh period is chosen by the organization. The Data Warehouse will be offline during this load period. As a rule, the Data Warehouse will not function when new data is being loaded into it. As the size of the source data grows, the refresh period also tends to grow.

Traditional Data Warehouses allowed decision making only on data that was last loaded into the Warehouse. This means that analysis was done on data that was either a week old. Business cannot respond to events quickly if the decision support system is not aware of the events. The longer the refresh period, the more the waiting period for the latest analytics. If a shorter refresh period is considered then the DWH will have to be in offline mode more often. The data lag is considerably reduced in a real-time Data Warehouse. Data lag here refers to the time difference between when the data enters the source system and when it reaches the Data Warehouse.

The ETL process is very important, yet a complex and time-consuming activity in the entire Data warehousing process [2]. This is because the data is adapted for the business by the ETL process. All the data in the Data Warehouse must undergo the ETL process. This focus of this paper is on using appropriate scheduling algorithms for ETL. An appropriate scheduling algorithm must be in place to ensure that the data arriving for ETL is appropriately handled and is loaded into the Data Warehouse consistently.

II. THE DATA WAREHOUSE ARCHITECTURE

Adapting to a real-time Data Warehouse means implementing the various Data Warehouse processes in real-time.

A typical Data Warehouse architecture consists of three layers [1].

A. The Staging Layer

This is where the data is stored prior to being scrubbed and transformed into the Data Warehouse. This layer acts as an integration layer for the data coming from different sources. The staging layer is mostly implemented as a separate database.

B. The Extraction Transformation and Loading (ETL) process

ETL processes constitute the backbone of a DW architecture, and hence, their performance and quality are of significant importance for the accuracy, operability, and usability of data warehouses [4].The function of the ETL process is to extract the data from the source, which can typically be one or more files or databases; transform the data by aggregating, consolidating, cleansing, applying business logic, cleansing, sorting, filtering etc,. ; And lastly loading this data into the Warehouse database. Since ETL ultimately determines the quality of the data that arrives for analysis, it is important to accurately perform ETL. But accuracy must not result in huge time consumption in real-time.

C. The Data Warehouse database

Once the data passes through the staging and the ETL layers, it can be loaded into the Data Warehouse. Business analysis happens on the data present in this database.

III. NEED FOR SCHEDULING

Scheduling algorithms are very important to any real-time system. Numerous scheduling algorithms are available and depending on the requirements, any one of the algorithms can be used. The choice of the algorithm is important in every real-time system and is greatly influenced by what kind of system the algorithm will serve [5]. A scheduler provides a policy for the execution of the various processes in the real-time system. Scheduler ensures that all the process are executed as per the priorities set. An online scheduler makes scheduling decisions based on the scheduling algorithm and the current state of the system.

IV. SCHEDULING ALGORITHMS

The below section provides an overview of the various scheduling policies that can be used for enabling real-time ETL.

A. Round Robin

The classical round robin scheduling can be implemented for ETL process [4]. The list of input tables where data has been recently updated in the staging database can be obtained by using the system tables. For ETL scheduling, it is assumed that push technology will be used to get data into the staging database from the source systems. In round robin scheduling, all the input tables are given some time slice to push the data into the ETL phase.

B. Memory Efficiency

The aim of this scheduling algorithm is to improve the efficiency of the memory. This algorithm selects the process that will consume the highest amount of data. This consumption data can be calculated from the below formula [3] -

Memory Gained is given by:

((Number of input tuples – Number of output tuples) / execution time) * Queue (1)

In Equation (1), input and output tuples refers to the number of input and output records. Execution time denotes the time taken by the process and Queue contains the remaining number of process in waiting.

C. Earliest Deadline First [2]

The deadline of each table is calculated here. The table which has the highest value will be updated first. The deadline is calculated as the time difference between the arrival of the latest set of data and the refresh time of the table that was refreshed the earliest among the tables in the Data Warehouse. EDF is a dynamic scheduling algorithm. Whenever a process arrives, the queue will be searched for the process that is closest to its deadline. This process will be scheduled next for execution.

D. Priority Scheduling

Some tasks might have higher priorities than others [2]. These priorities can be because they are more important to the business or because they might update a large number of end reports. If the user wishes to set priorities to tables, then priority based scheduling algorithm can be implemented. As data comes in, the priority queue is checked and depending on the assigned priorities, the data is processed.

E. Max-Benefit [3]

In the max-benefit algorithm, the updation of tables that provide the maximum benefit is processed first. As an example consider that the updation of table A results in 5 queries or reports at the decision making end being refreshed. The updation of table B will result in 3 reports being refreshed. The Max-Benefit algorithm will update table A first.

V. Results and Discussions

Depending on the type of Data Warehouse deployed, a suitable hard or soft real-time scheduling algorithm can be selected. Hard real time systems have tight deadlines and should strictly abide by the deadline. In the case of a hard real time system, if a system misses the deadline then the system is a failure. So EDF is optimal for hard real time systems. If delayed decision making results in any substantial loss in the Data Warehouse, then EDF can be used for scheduling.

However for soft real time systems, the lateness is the performance criteria. As long as data don’t arrive too late, the system performs well. For such systems, the max-benefit algorithm performs well as majority data will be updated. For Systems with memory constraint, the memory efficiency algorithm can be used. If the users have prioritized reports then the priority scheduling algorithm must be used.

VI. Conclusion and Future Work

Scheduling updates non-preemptively in the Data Warehouse has been discussed in this work. As a future work, new algorithms that can perform better than EDF and Max-Benefit can be determined. Also in this work, the number of updates for a single table is not considered. All the updates for a single table is considered as a single job. If a single table has multiple updates, then this might cause a queue. As a future enhancement, the number of updates a table should have at a time can be considered.

REFERENCES

[1] Ralph Kimball, Margy Ross , The Data Warehouse Toolkit , Wiley , 2002.

[2] Lucas Golab, Theodere Johnson and Vladislav Shkapenyuk ,Scalable, Scheduling of Updates in Streaming Data Warehouse, IEEE Transactions on knowledge and data engineering ,Vol. 24, N0. 6, JUNE 2012.

[3] Anastasios Karagiannis , Anastasios Karagiannis and Alkis Simitsis , Scheduling strategies for efficient ETL execution , Information Systems 38 (2013) 927–945.

[4] Oracle9i Data Warehousing Guide Release 2 (9.2), Part Number A96520-01. Available at: http://docs.oracle.com/cd/B10501_01/server.920/a96520/ extract.ht

[5] Rajib Mall , Real-Time Systems - Theory and Practice , Pearson Education, May 2009.