Monitoring Business Transactions for a Real-time Data Warehouses

Abdelmgeid A. Ali
Faculty of Science,
Computer Science Department,
Minia University,
Minia, Egypt

Waleed M. Mohamed
Faculty of Computers and Information,
Computer Science Department,
Minia University,
Minia, Egypt

ABSTRACT
Real-time business intelligence (RTPI) is an approach to data analytics that enables business users to get up-to-the-minute data by directly accessing operational systems or feeding business transactions into a real-time data warehouse and business intelligence (BI) system. Business Intelligence (BI) technology has helped many organizations to make better and faster decisions and improve its performance. RTBI allows organizations to evaluate business processes and take strategic action on the current overall business environment. The ability to manage and effectively present the volume of data tracked in today's business is the cornerstone of data warehousing, but when a business users require up-to-date or real-time data for the purpose of analysis, which presuppose the building of a real-time data warehouse (RTDW). In this paper we propose a real-time framework to support this up-to-date process. Our framework is based on reading transaction log file of external data sources to determine that data changed using changed data capture, then load this data to data warehouse. This framework minimizes impact to the source system and the target data warehouse system.

Keywords
Change Data Capture (CDC), Extract, Transform and Load (ETL), Replicate, Extract, Transform and Load (RETL), Real-Time Data Warehouse (RTDW)

1. INTRODUCTION
Data warehousing is one of the more powerful tools available to support a business enterprise, it provides historical data for analytical processing, decision making and for data mining tools. A Data Warehouse extracts data from multiple heterogeneous operational data sources OLTP (On-Line Transaction Processing) and stores summarized integrated business data in a central repository used by analytical applications OLAP (On-Line Analytical Processing) with different user requirements.
The ETL processes are responsible for identifying and extracting the needed data from the OLTP data sources, transforming this data into a target table format, cleaning the data and conforming it into an adequate integrated format for updating the data area of the Data Warehouse and, finally, loading the final formatted data into its Data Warehouse [1].
Almost immediately after the original data is committed, that data moves direct from the originating publisher to the data warehouse. Both the before and after image of a record is available in the data warehouse memory, thereby supporting easy and efficient processing for query and analysis at any time.
A real-time data warehouse has traditionally been done in an offline fashion. This meant that while updating the data warehouse, OLAP applications could not access any data. Since data warehouses are currently the backbone of decision support and Business Intelligence systems [2]. Needing real-time data warehouse depending on the real-time in operational data store systems that data is require in data warehouse system, some of the applications for real time data are [3]:

- Most recent information is required to detect suspicious group of passengers in airlines.
- In banking systems real time data is required for certain critical areas of operations such as auditing systems and anti money laundering.
- To detect ATM frauds also we need real time data.

Given the benefits of real-time data warehousing, it is difficult to understand why the "snapshot" copy process has prevailed. Currently, the dominant method of replenishing data warehouses and data marts is to use extraction, transformation and load tools that "pull" data from source systems periodically at the end of a day, week, or month and provide a "snapshot" of your business data at a given moment in time. The batched data is then loaded into a data warehouse tables. During each cycle, the warehouse table is completely refreshed and the process is repeated no matter whether the data has changed or not.

As reliability aspect of a real time data warehouse, the decision support system is only dependent on the source systems to be available at real time. Real time architecture demands continuous uptime and reliance on the real time nature of the reporting and business intelligence systems will require much more robust infrastructure such as the services provided by network systems management. Real Time Data Warehouse architecture consists of a lot of views, approaches. Before building RTDW, requirements must be clearly analyzed from a real time needs perspective. Some of these requirements are presented in [4].

2. REAL-TIME DATA WAREHOUSE FRAMEWORK
The proposed framework based on real-time data warehouse architecture, as shown in Fig.1.

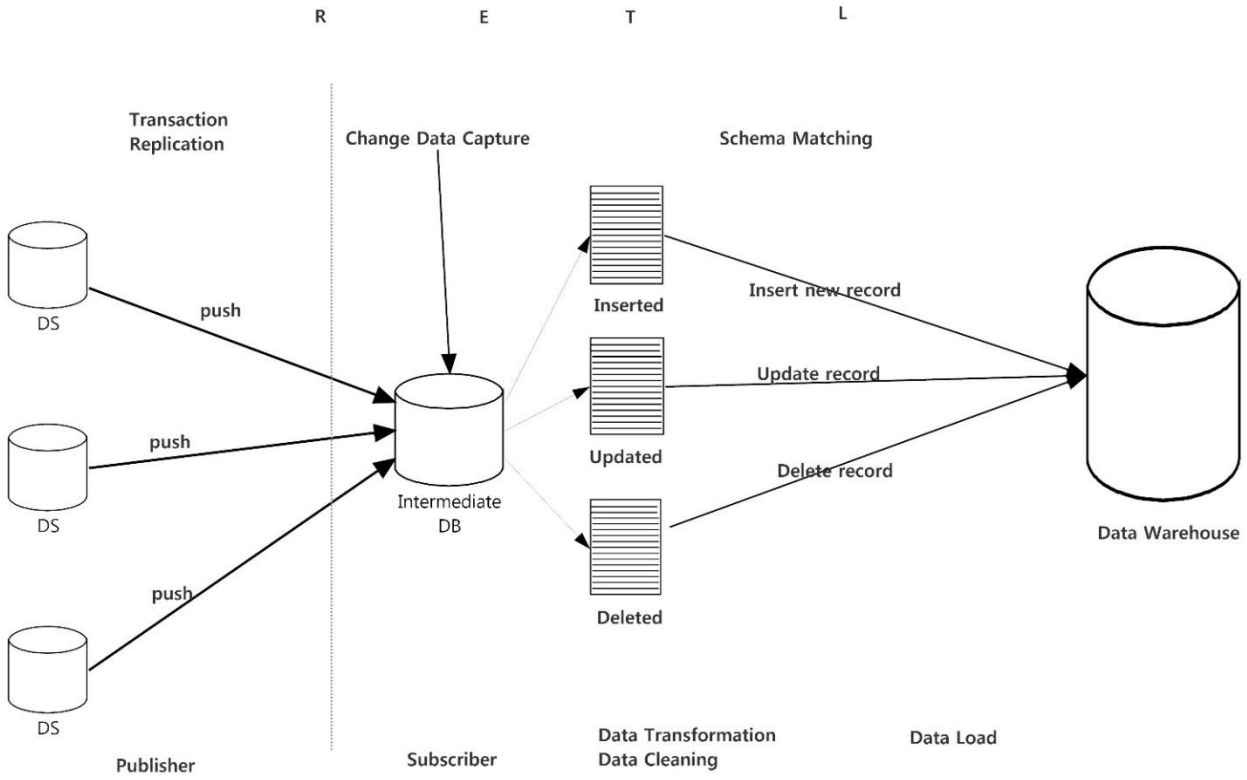

Fig.1. System architecture for real-time data warehousing (RTDW)

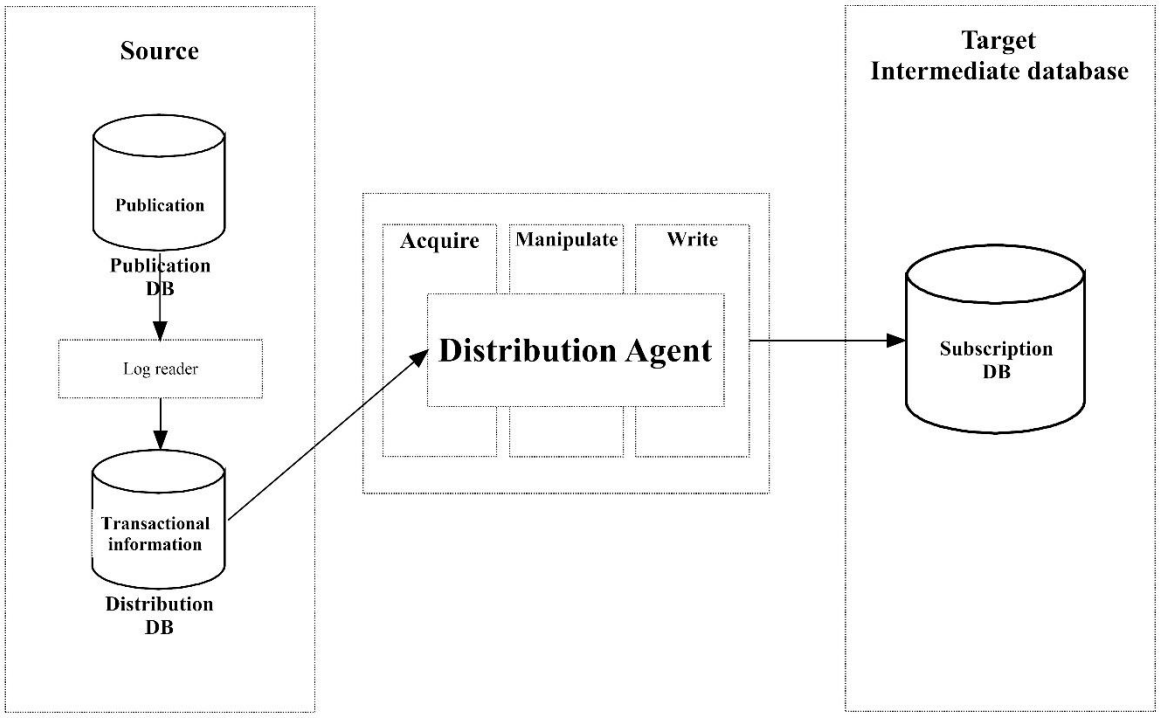

Fig.2. Transaction replication between data source and intermediate database

Components of this architecture as follows:

- Transaction Replication
- Intermediate Database
- Change Data Captures
- Schema Matching
- Data Transformation and Cleaning
- Data Load

A. Transactional replication

In transactional replication, each committed transaction in OLAP is replicated to the subscriber (intermediate database) as it occurs. You can control the replication process so that it

will accumulate transactions and send them at timed intervals, or transmit all changes as they occur depending on your business need. Transactional replication begins with a snapshot that sets up the initial copy, that copy is updated by the copied transactions as shown in Fig.2. Once the initial snapshot has been copied, transactional replication uses the log reader agent to read the transaction log of the published database and stores new transactions in the distribution Database. The distribution agent then transfers the transactions from the publisher (OLTP) to the subscriber.

Also, in server-to-server environments has been used for following situations:

- Need incremental changes to be propagated from external data sources (publisher) to intermediate database (subscriber).
- Need near real time data with low latency.
- Publisher has a very high volume of insert, update and delete activity.
- Can be used for non-sql server database also.

B. Intermediate Database

Is the same structure as the target data warehouse.

C. Change data captures (CDC)

There are different ways to implement CDC. Some of these techniques are following:

1) Database Triggers: a trigger is a special kind of stored procedure written by user that is execute whenever an attempt is made to insert, update and delete the data in the table against which this trigger is written [5]. Disadvantage of using triggers for CDC is that it puts overhead burden on the source database and impacts the performance of the system.

2) Timestamps: Each record in source database contains a field that stores the date and time when the data was inserted and updated. CDC reads this field and captures the most recent modified data [6]. Disadvantage of using this CDC approach is that cannot capture deleted record because the field that contain the date and time is deleted with the record, and also, it cannot capture all updates attempted to each record because the available is the last update attempted (net change).

3) Log file: Database maintains log files to minimize the loss of data in case of system disaster. Speeds up the diagnostic process. Log files contain detail of all the transactions which has taken place on OLTP tables such as any insert, update and delete transactions in the database. Only read log file is one approach through which all changes can be captured and updated into data warehouse and also log-based technique has minimal impact on source database.

4) Partitioning: Some source systems might use range partitioning, such that the source tables are partitioned along a date key, which allows for easy identification of new data. For example, if you are extracting from an orders table, and the orders table is partitioned by week, then it is easy to identify the current week's data.

Due to the previous advantages of reading log file technique, it will use as transaction replication with CDC technique in this paper. This feature is applying in intermediate database, after applying this feature, three tables are created in this database (inserted, updated and deleted tables).

CDC has been applied on the intermediate database not on the source system (OLTP) because CDC tracks insert, update and delete in the transaction log, and insert the changes into changing tables that associated with the tracked tables. The change tables are automatically created in the same database, for every change in OLTP database, CDC does another insert in the same change table in the same database, which can have a noticeable performance hit on OLTP database.

D. Schema matching

Is a manipulation process on schemas that takes two heterogeneous schemas (possibly have auxiliary information) as input and produces as output a set of mappings that identify semantically relation between elements of the two schemas. map the XML Schema of intermediate database (source data) and the star schema of data warehouse tables (target data) to identify semantic correspondences between them. These correspondences are explicitly described as mapping rules in our work [7].

E. Data transformation and cleaning

The data transformation process typically consists of multiple steps where each step may perform schema- and instance-related transformation (mappings) [8]. In metadata and data warehouse, a data transformation converts a set of data values from the data format of a source schema into the data format of a destination schema. Data cleaning, also called data cleansing or scrubbing, deals with detecting and removing errors and inconsistencies from data in order to improve the quality of data [7].

F. Data load

After capture the changed data (insert, update and delete) and applying the transformation and cleaning, then this changed data loaded into the data warehouse system. Truncate the associated table that created in intermediate database to keep the intermediate database with minimum impact.

If the operation updates the data in fact table, then no problem occurs when applying data warehouse analysis, but if the operation updates the data in dimension table such updating customer address, may be problem occur, where dimensions provide description or meaning for fact table data. Some dimension data may change over time, so the historical accuracy is changed. To keep the DW system accuracy, so it considered when designing real-time data warehouse system. Data is not overwriting in the data warehouse, but a new row for this dimension table must be inserted, usually creates a primary key problem. So it must be adding a surrogate key (data warehouse key) that uniquely identifies every row in the dimension table, and add another column or two to flag the current value and provide date/time perspective.

3. A REAL-TIME DATA WAREHOUSE ALGORITHM

Algorithm: RTDW

1. Begin
2. Create an intermediate DB.
3. Load data from data sources to intermediate DB and data warehouse (ETL).

4. If data in dimension table updated then
   Add a surrogate key column (DW key) in dimension table
   Add flag column the current value

// To perform incremental load:

5. Implement transaction replication form data sources (publisher) to intermediate DB (subscriber).
6. Monitoring a change in the data (intermediate DB) .
7. Partition the changed data into three tables according to insert data, update data and delete data.
8. If monitor changed data = delete, then
   Put this data into deleted table
   Delete this from DW
   Truncate deleted table
9. Else If monitor changed data = insert then
   Put this data into inserted table
   Apply data transformation and cleaning using schema matching
   Insert this data into DW
   Truncate inserted table
10. Else monitor changed data = update then
    Put the last update data for the same row into updated table
    Apply data transformation and cleaning using schema matching
11. If update belong dimension table then
    Set the current flag column =0 in DW dimension table
    Insert new row in DW dimension table, set flag=1
12. If update belong fact table then
    Update DW
    Truncate updated table

End

4. CONCLUSION

A transaction replication based on real time data warehouse has been designed, developed and built. A transaction replication is important component as the publish stage that needed data from the OLTP system to the intermediate database. Capturing change data from source systems is also a major problem for data warehouse constructions. Our framework monitors the changed data in intermediate database then load it into data warehouse system. This framework minimizes impact to the source system because no additional tables or queries to the source database are required to support the data capture process. The capture module reads once in intermediate database, and then immediately moves the captured data to the data warehouse tables.

5. REFERENCES

[1] Sanjay, K Reddy V. Mallikarjuna Jena. "Active Data Warehouse Loading by Tool Based ETL Procedure." International Conference on Information and Knowledge Engineering, 2010: 196-201.

[2] Kobielus, j. "The Forrester Wave: Enterprise Data Warehousing Platforms." Forrester Research, Q1, 2009.

[3] Tanvi Jain, Rajasree S, Shivani Saluja. "Refreshing Data warehouse in Near Real-Time." International Journal of Computer Applications 46, no. 18 (May 2012): 24-28.

[4] R., Cacerta, J. Kimball. The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data. Wiley, 2004.

[5] C. R. Valencio, M. H. Marioto, G. F. Zafalon, J. M. Machado. "Real Time Delta Extraction Based on Triggers to Support Data Warehousing." The International Conference on Parallel and Distributed Computing, Application, and Technologies (PDCAT). 2013.

[6] R. Kimball, J. Caserta. The Data Warehouse ETL Toolkit:Practical Techniques for Extracting, Cleaning. John Wiley & Sons, 2004.

[7] Abdelmgeid A. Ali, Tarek A. Abdelrahman, Waleed M. Mohamed. "Using Schema Matching in Data Transformation For Warehousing Web Data." International Journal of Information Technologies and Knowledge 7, no. 3 (2013): 230-240.

[8] Erhard Rahm, Hong Hai Do. "Data Cleaning: Problems and Current Approaches." IEEE Data Engineering Bulletin 23, no. 4 (2000): 3-13.