Framework Of Change Data Capture And Real Time Data Warehouse

Mayuri B. Bokade, Prof. S. S.Dhande, Prof. H. R. Vyavahare

1. P. G Scholar,
Sipna Collage of Engineering and Technology, Amravati
Department of Computer Science and Engineering,
Sant Gadge Baba Amravati University, Maharashtra, India

2. Associate Professor,
Sipna Collage of Engineering and Technology, Amravati
Department of Computer Science and Engineering,
Sant Gadge Baba Amravati University, Maharashtra, India

3. Assistant Professor,
Sipna Collage of Engineering and Technology, Amravati
Department of Computer Science and Engineering,
Sant Gadge Baba Amravati University, Maharashtra, India

Abstract

This paper proposes a framework of change data capture and data extraction in real time data warehouse. Data capture framework supports to the trigger, data replication, and other capture methods. Real-time data warehouse describes a system that reflects the business in real time and it proposes Real-time Scheduling Strategies. In large organization, huge amounts of data are generated and consumed for decision makers that need up-to-date information every time for processing data. Real time data warehouse, which will manage the ETL process with a more compact data and a shorter period is needed. In future, this paper will improve the quality of data using real-time change data capture (CDC) techniques. A change data capture framework has a very important role in ETL design for Data Warehouses.

Keywords

Change Data Capture, Real time Data Warehouse, ETL

1. Introduction

Traditionally data warehouses do not contain today's data i.e. up to date data. They are usually loaded with data from operational systems. Expectations of accurate and trusted information to be delivered to the right people, at the right time, and in the right format is important regardless of geographical location or industry of business. A real-time data warehouse is required because of the lack of real-time update in traditional data warehouse. Real-time data warehousing typically describes a system that reflects the business in real time. When we have to take decisions very quickly, access to real-time data is one of the key considerations in almost every corporation. Taking strategic decisions based on out-of-date data can produce wrong results. As today's decisions in the business world become more real-time, the systems that support those decisions need to keep up. Data Warehouse, Business Intelligence, Decision Support systems quickly begin to incorporate real-time data. Keeping data current from when data is captured until it is available to decision makers in this context is a difficult task. Change Data Capture (CDC) is an innovative mechanism to data integration, based on the identification, capture, and delivery of changes made to enterprise/ operational/transactional data sources. By

processing only the changes, CDC makes the data integration process more efficient and real time and reduces its cost by reducing the latency between the time a change in the data occurs and the time the same change is made available to the business user. CDC uses ETL (Extract, Transform, and Load) tool to process DW updates. ETL is a software program that extracts the data from the source system, transforms and cleanses the data, and then loads it into the DW. This paper proposes a framework of change data capture and data extraction in real time data warehouse. This paper focuses on how data is extracted, how data is transformed and loaded into the data warehouse.

2. Change Data Capture:

The ability to detect the changed data in source systems and capture these changes is called **Change Data Capture** (**CDC**).Handling those changes is the most difficult and challenging task. Aim is just detect the data that have been changed since the last load; implementation of CDC can cause many problems and questions. Technically we can say that Change Data Capture (CDC) is the process used to achieve real-time synchronization. Change Data Capture quickly identifies and processes only the data that has changed and makes the changed data available for further use.

An objective of CDC is to improve efficiency by processing the minimum amount of data. CDC solutions can provide filters that allow reducing the amount of information and delivering only the relevant records. Such filters enable the delivery of records based on the type of change (i.e. inserts, updates), deliver records only when a change happened to specific fields, or specify a subset of fields from the original record that are required for processing. These filters can achieve maximum benefit when they run close to the source, by reducing the amount of records that traverse the network.

Given the need of many companies for up-to-the-minute information, they have started looking for ways to update their DW in real-time, considerably reducing latency.

CDC provides an efficient mechanism to keep an ODS up-to-date, by identifying and delivering the changes on a continuous basis, rather than periodically querying the entire data base for changes. In addition, CDC can push changes in near real-time to support ODS applications that have very low latency requirements.

With CDC, the process of propagating data that is having one or more copies of the data from a given data source, can be made much more efficient and reduce the latency in making new data available.

Algorithm 1. Change data capture

```
1 BEGIN
2 EXEC SQL EXECUTE
3 begin
4 dbms_logmnr_d.build(dic_filename=>'vdict.ora');
5 dbms_logmnr.add_logfile(LogFileName=>'redo01.  log',Options=>dbms_logmnr.new);
6 dbms_logmnr.add_logfile(LogFileName=>'redo02. log',Options=>dbms_logmnr.addfile);
7 dbms_logmnr.add_logfile(LogFileName=>'redo03. log',Options=>dbms_logmnr.addfile);
8 dbms_logmnr.start_logmnr(DictFileName=>'vdict. ora');
9 end;
10 END-EXEC;
11EXEC sql select max(scn) into :scn_now
   fromv$logmnr_contentswhere timestamp in (Specified Time);
12 EXEC sql select max(taskid) into :taskid from test.task;
13 EXEC sql select max(scn_old) into :scn_old from   scn1;
14 if(scn_now == scn_old)
15 return;
16 else
17 Extract the analyzed dictionary information and store the incremental data;
18 END
```

In the above algorithm1of change data capture, the steps from 2 to10 set the data dictionary and load the log data for analyzing; the steps from 11to17 analyze the log file, capture changed data information and export changed data for further processing.

3. CDC Techniques

There are several techniques and technologies for handling the change data capture processes (CDC process), some of them are:

3.1 Transaction log file

Almost all database management systems have a transaction log file that records all changes and modifications in database made by each transaction. To capture changes made to the database we can scan and analyze the contents of the database transaction log. When change data capture process use this technique then transaction log analyzing does not affect the operational transactional database.

3.2 Trigger method

One of the best techniques for CDC is adding triggers to tables whose changes should be controlled and managed. Triggers can be created in operational systems to keep track of recently updated records. They can then be used in conjunction with timestamp columns to identify the exact time and date when a given row was last modified. We do this by creating a trigger on each source table that requires change data capture. Following each DML statement that is executed on the source table, this trigger updates the timestamp column with the current time. Thus, the timestamp column provides the exact time and date when a given row was last modified. In this method, triggers are turned on when an operation of insert, update or delete happens and record the information about the changes to specific tables or files but obstacles occurs while implementing additional triggers.

3.3 RDBMS Replication

In the process of DBMS replication copies of data are automatically distributing between few database servers, and keeping the distributed information synchronized. If RDBMS supports replication, we can also use this mechanism to replicate data to a data warehouse. Replication is not exactly the change data capture mechanism, so this requires additional efforts to use replication to achieve the CDC needs. Also, because of incompatibility replication may not be the good solution in case of heterogeneous source systems.

4. Change Data Capture Framework

The change data capture framework support to the trigger, data replication, and other capture methods. Figure1 shows the framework of change data capture based on transaction log files.

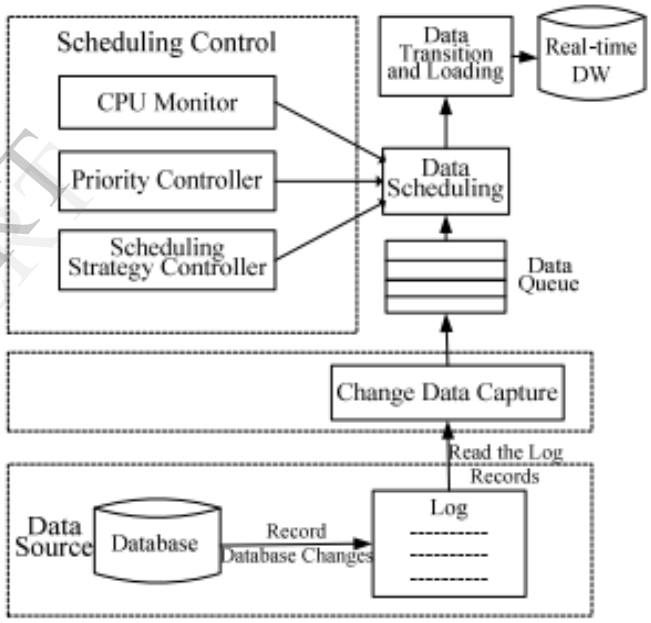

Figure 1: Framework of Change Data Capture Based On Transaction Log

This Framework includes change data capture component based on log files, schedule control and data transition and loading, scheduling component and other components. Source database will continue to generate new data and Log file records the database changes, but the importance of all data is different. The important data have the priority to be imported to

RTDWH. Data queue is composed of data that are imported by change data capture component.Scheduling component uses the First in First Serve scheduling (FIFS) and the priority scheduling, and it also ensures freshness of data and quality of data in real time data warehouse. At the same time scheduling controller monitors utilization rate of CPU, using real-time feedback method to adjust scheduling strategy and to provide better data quality assurance. Scheduling controller solves the task scheduling problems, while addressing priority scheduling problem for the important task. It sets the priority of all tasks. Priority controller classifies the tasks according to the priority and delivers tasks to a different running queue.Data transition and loading component, firstly cleans data and then transformed it into specific format and then connect to RTDWH by loading component.

5. ETL and Data Warehousing (DW)

Reducing the cost and resources associated with updating an enterprise DW is the problem for business or IT. CDC in working with ETL tools provides a new approach to moving information into a DW and reduces cost and resources associated with updating an enterprise DW. Traditionally, DW updates are processed with an ETL (Extract, Transform, and Load) tool. ETL is a software program that extracts the data from the source system, transforms and cleanses the data, and then loads it into the DW.

These processes require that the operational system(s) be put off line for a given period of time. This period of time is referred to as a "Batch Window", typically measured in hours and sometimes days, during which the system is busy with moving the data and cannot perform operational and other mission critical functions. Given the limitation of this 'bulk' approach, most IT shops update their DW only daily, and often on a weekly basis. Given the need of many companies for up-to-the-minute information, they have started looking for ways to update their DW in real-time, considerably reducing latency. For example, in case of Share Market ,when data is unavailable to a Business Analyst then result can be millions of dollars in losses for each hour of unavailability.CDC provides a new approach to moving information into a DW and can work with ETL.

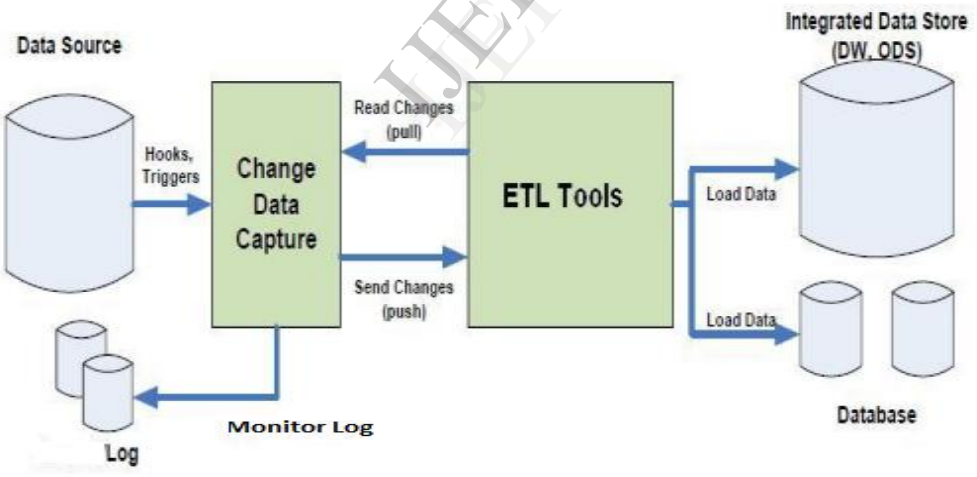

Figure 2: Working of CDC with ETL

CDC delivers changes to an ETL tool in batch or real-time, allowing to dramatically improve the efficiency of the entire process, reduce or totally eliminate batch windows, deliver information in low latency, and reduce the associated costs including CPU cycles, storage, network bandwidth and human resources.

6. ETL Process

Extract, Transform and Load (ETL) refers to a process that involves: extracting data from outside sources, transforming it to meet quality and operational needs, and then loading that data into the target i.e. database or data warehouse.ETL process pulls out data from various sources and load it to a data warehouse.ETL systems are used to move data from one data warehouse to another data warehouse. ETL process first extract data from various data sources or operational applications then transforms values of inconsistent data, cleanses "bad" data, filters it and then load that data to target data warehouse.

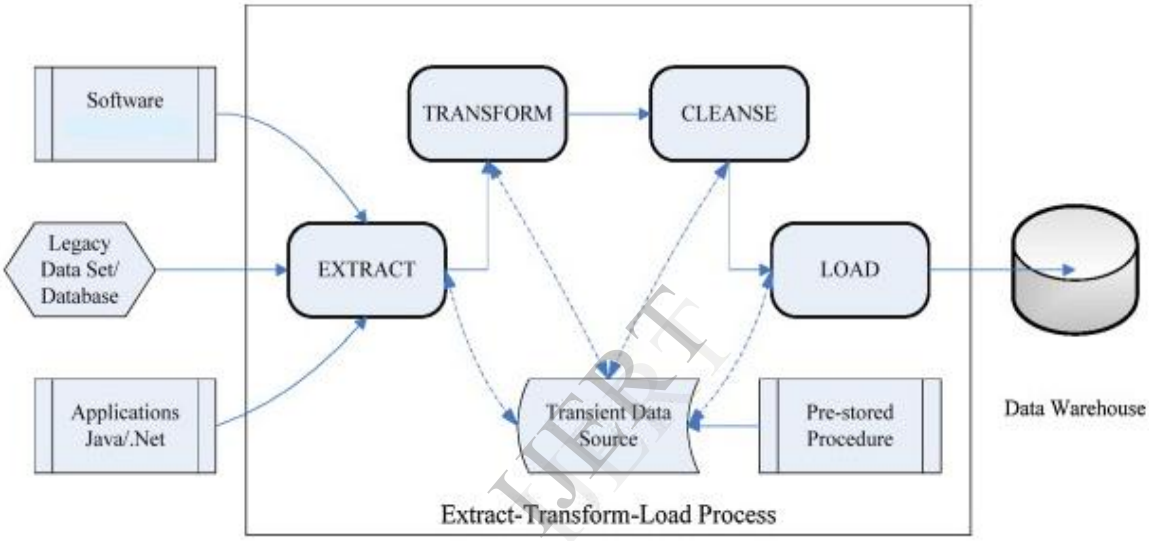

Figure3: Extract-Transform-Load Process.

The ETL process is not a one-time event; new data is added to a data warehouse periodically. Typical periodicity may be monthly, weekly, daily, or even hourly, depending on the purpose of the data warehouse and the type of business it serves.

7. Extraction in Data Warehouse

Extraction is the operation of extracting data from a source system for further use in a data warehouse environment. This is the first step of the ETL process. After the extraction, this data can be transformed and loaded into the data warehouse. Designing and creating the extraction process is often one of the most time-consuming tasks in the ETL process and in the entire data warehousing process. The source systems might be very complex and poorly documented, and thus determining which data needs to be extracted can be difficult. The data has to be extracted normally not only once, but several times in a periodic manner to supply all changed data to the data warehouse and keep it up-to-date. Moreover, the source system typically cannot be modified, nor can its performance or availability be adjusted, to accommodate the needs of the data warehouse extraction process.

There are various Extraction Methods in Data Warehouses:

7.1Logical Extraction Methods

There are two types of logical extraction:

7.1.1 Full Extraction

The data is extracted completely from the source system. Because this extraction reflects all the data currently available on the source system, there's no need to keep track of changes to the data source since the last successful extraction. An example for a full extraction may be an export file of a distinct table or a remote SQL statement scanning the complete source table.

7.1.2 Incremental Extraction

At a specific point in time, only the data that has changed since a well-defined event back in history will be extracted. To identify this change there must be a possibility to identify all the changed information since this specific time event. This information can be either provided by the source data itself such as an application column, reflecting the last-changed timestamp or a change table where an appropriate additional mechanism keeps track of the changes besides the originating transactions.

7.2 Physical Extraction Methods

The data can either be extracted online from the source system or from an offline structure.

There are the following methods of physical extraction:

7.2.1 Online Extraction

The data is extracted directly from the source system itself. The extraction process can connect directly to the source system to access the source tables themselves or to an intermediate system that stores the data in a preconfigured manner (for example, snapshot logs or change tables). With online extractions, you need to consider whether the distributed transactions are using original source objects or prepared source objects.

7.2.2 Offline Extraction

The data is not extracted directly from the source system but is staged explicitly outside the original source system. The data already has an existing structure (for example, redo logs, archive logs or transportable table spaces) or was created by an extraction routine.

8. Transportation in Data Warehouse

Transportation is the operation of moving data from one system to another system. In a data warehouse environment, the most common requirements for transportation are in moving data from one data warehouse to another data warehouse or data mart. Transportation is one of the simple stage of the ETL process, and can be integrated with other stages of the process.

There are three basic Transportation mechanisms for transporting data in warehouses:

8.1 Transportation Using Flat Files

The most common method for transporting data is by the transfer of flat files, using mechanisms such as FTP or other remote file system access protocols. Data is unloaded or exported from the source system into flat files and is then transported to the target platform using FTP or similar mechanisms .Because source systems and data warehouses often use different operating systems and database systems, using flat files is often the simplest way to exchange data between heterogeneous systems with minimal transformations. However, even when transporting data between homogeneous systems, flat files are often the most efficient and most easy-to-manage mechanism for data transfer.

8.2Transportation Through Distributed Operations

Distributed queries, either with or without gateways, can be an effective mechanism for extracting data. These mechanisms also transport the data directly to the target systems, thus providing both extraction and transformation in a single step. The success or failure of the transportation is recognized immediately with the result of the distributed query or transaction.

8.3Transportation Using Transportable Tablespaces

Oracle transportable tablespaces are the fastest way for moving large volumes of data between two

Oracle databases. Previous to the introduction of transportable tablespaces, the most scalable data transportation mechanisms relied on moving flat files containing raw data. These mechanisms required that data be unloaded or exported into files from the source database. Then, after transportation, these files were loaded or imported into the target database. Transportable tablespaces entirely bypass the unload and reload steps. Using transportable tablespaces, Oracle data files (containing table data, indexes, and almost every other Oracle database object) can be directly transported from one database to another. Furthermore, like import and export, transportable tablespaces provide a mechanism for transporting metadata in addition to transporting data. Transportable tablespaces have some limitations: source and target systems must run on the same operating system.

9. Loading in Data Warehouse

ETL is the ideal solution for the loading of large volumes of data and also offers advanced transformation capabilities. After the data has been cleansed and transformed into a structure consistent with the data warehouse requirements, data is ready for loading into the data warehouse. You may make some final transformation during the loading operation, although you should complete any transformations that could identify inconsistencies before the final loading operation. You can load the data to data warehouse using various mechanisms such as SQL loader or External Tables or Export /Import .The load phase loads the data into the end target, usually the data warehouse (DW). Depending on the requirements of the organization, this process varies widely. Some data warehouses may overwrite existing information with cumulative information, frequently updating extract data is done on daily, weekly or monthly basis. Other DW (or even other parts of the same DW) may add new data in a historical form, for example, hourly. To understand this, consider a DW that is required to maintain sales records of the last year. Then, the DW will overwrite any data that is older than a year with newer data. However, the entry of data for any one year window will be made in a historical manner. The timing and scope to replace or append are strategic design choices dependent on the time available and the business needs. More complex systems can maintain a history and audit trails of all changes to the data loaded in the DW.As the load phase interacts with a database, the constraints defined in the database schema — as well as in triggers activated upon data load — apply (for example, uniqueness, referential integrity, mandatory fields), which also contribute to the overall data quality performance of the ETL process.

10. Real Time Data Integration

Next generation Data Integration and ETL (Extract-Transform-Load) tools need to support Change Data Capture (CDC) and implementing CDC makes data and information integration in REAL TIME significantly more efficient, and delivers data at the right-time. Change data capture is an approach to data integration based on the identification, capture, and delivery of only the changes made to operational/transactional data systems. By processing only the changes, CDC makes the data integration, and more specifically the 'Extract' part of the ETL process more efficient. When done correctly, it also reduces the 'latency' between the time a change occurs in the source systems and the time the same change is made available to the business user in the data warehouse. Change data capture is a technique that can be used to address many of the data management or data integration challenges that IT managers are faced with. These challenges range from addressing batch window challenges to providing live data feeds to support a workflow within a business application.In the next generation implementing

CDC makes data and information integration in real-time significantly more efficient, and delivers data at the right-time. When business requirements are for only certain changes to be captured, then transferring all changes it would be wasteful. The most advanced CDC solutions therefore provide filters that reduce the amount of information transferred, again minimizing resource requirements and maximizing speed and efficiency.

11. Conclusion and Further Work

This paper proposes a framework of change data capture based on transaction log in real time data warehouse. We explain various techniques of change data capture. Change data capture in working with ETL, successfully provides new approach to move information into data warehouse and also makes information integration more efficient in real time i.e. delivers information at the right-time. We introduce

ETL and CDC using only Oracle database but we can research it with other database. In future we can achieve real-time data delivery by combining strengths of Change Data Capture (CDC) and ETL tools.

12. References

[1] Michael Haisten. *Real Time Data Warehouse*: *The Next Stage in Data Warehouse Evolution.* DM Review, 2003.

[2] K. D. Kang, S. Son, J. Stankovic, T. Abdelzaher. *A QoSSensitive Approach for Timeliness and Freshness Guarantees in Real-Time Databases.* In the 14th Euromicro Conference on Real-Time Systems. 2002, 203-212.

[3] Lin Ziyu, Yang Dongqing, Song Guojie. *Study on change data capture in Real-time data warehouse*. Journal of Computer Research and Development, 2007, 44: 447-451.

[4] Itamar Ankorion. *Change data capture-efficient ETL for real-time* BI[J], DM Review, 2005, 16(1):23-27.

[5]Javed, Dr.Muhammad Younus. , Nawaz, Asim. , 2010. *Data Load Distribution by Semi Real Time Data Warehouse, In: Computer and Network Technology (ICCNT),* 2010 Second International Conference On page(s): 556 - 560

[6] Inmon, W.H. 2005. *Building the Data Warehouse Fourth Edition.Canada* : Wiley Publishing.Inc.

[7] Simitsis, A.; Vassiliadis, P.; Sellis, T.; *Optimizing ETL Processes in Data Warehouses.In Data Engineering,* 2005. ICDE 2005.Proceedings. 21st International Conference on Digital Object, Page(s): 564 – 575

[8] Vandermay, John., 2001. *Considerations for Building a Real-time Data Warehouse*

[9] Langseth ,Justin., 2004, *Real-Time Data Warehousing: Challenges and Solutions.*

[10] Jie Song; Yubin Bao; Jingang Shi; 2010, *A Triggering and Scheduling Approach for ETL.Computer and Information Technology (CIT),* 2010 IEEE 10th International Conference on, Page(s): 91 – 98.

[11] Attunity Ltd , 2009, *Efficient and Real Time Data Integration With Change Data Capture,* Tersedia di http://www.attunity.com/cdc_for_etl .

[12] Jingang Shi, Yubin Bao, Fangling Leng, Ge Yu.2008,*Study on Log-Based Change Data Capture and Handling Mechanism in Real-Time Data Warehouse. In International Conference on Computer Science and Software Engineering*, CSSE 2008, Volume 4: Embedded Programming / Database Technology / Neural Networks and Applications / Other Applications, December 12-14, 2008, Wuhan, China. pages 478-481, IEEE Computer Society, 2008.

[13] International Journal of Computer Theory and Engineering, Vol. 2, No. 5, October, 2010 1793-*8201 Empirical Study on Dynamic Warehousing* C K Bhensdadia*, Yogeshwar P Kosta.

[14] R. Kimball and J. Caserta, *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting*, Cleanin. John Wiley & Sons, 2004.

[15] 2008 International Conference on Computer Science and Software Engineering *Study on Log-Based Change Data Capture and Handling Mechanism in Real-Time Data Warehouse*
JinGang Shi, YuBin Bao, FangLing Leng, Ge Yu Department of Computer Science, Northeastern University,Shenyang,
Chinashijg5@163.com,baoyubin,lengfangling, yuge}@mail.neu.edu.cn.

[16] International Journal of Computer Theory and Engineering, Vol. 2,
No. 5, October, 2010 1793-*8201Empirical Study on Dynamic Warehousing*, C K Bhensdadia, Yogeshwar P Kosta.

[17] A Strategic Component of Data Integration Copyright © 2006
Attunity Ltd. All rights reserved *Real Time Business Intelligence*
Enabling Effective Decision Making Strategic, Real Time Data
Integration Platform With Change Data Capture.