ISSN XXXX XXXX © 2019 IJESC

Research Article Volume 9 Issue No.2

Survey on Data Warehouse from Traditional to Real Time and Society Impact of Real Time Data

Farhad Alam[1], Neel Kamal[2]
Production Analyst[1], Research Scholar[2]
eBay (Shanghai), China[1]
Himalayan University, Arunachal Pradesh, India[2]

Abstract:
The Traditional data warehouse does not contain today data. They are usually loaded with data from operational systems at most weekly or in some cases nightly. It is hard to retrieve these data and treat them. As today's decisions in the business world become more real-time, the systems that support those decisions need to keep up. It is only natural that Data Warehouse, Business Intelligence, Decision Support, and OLAP systems quickly begin to incorporate real-time data. In this paper, we are interested in giving a survey on data warehousing starting from a traditional data warehouse to a real time data warehouse and what is the society impact of real time data. This survey, focus on data warehouse architecture. It details the changes in the Extract-Transform-Load process to deal with real time data warehousing. It sketches the integration data in the real time data warehouse. Finally, a comparative study concerning the real data warehouse approaching is also presented in this paper.

Keywords: Traditional Data Warehouse, Real Time Data Warehouse, Near Real Time Extract Transform Load, Real Time Data Society Impact.

I. INTRODUCTION

A Data Warehouse is a central repository of integrated data from more disparate sources. It stores historical data to create analytical reports for knowledge workers throughout the enterprise. A DWH includes a server, which stores the historical data and a client for analysis and reporting. these data are turned into structured information that can easily be handled by decision making processes used to make strategic decisions by means of Online Analytical Processing (OLAP) techniques. Because of the new business requirements forever up-to-second updated information and since timely data ensure better-informed decisions, real-time data warehousing is one of the savior trends that provide an access to accurate, integrated, consolidated view of the organizations information in real-time [8]. For a real time data warehouse, no official definition currently exists in the literature. A Real Time Data Warehouse (RTDW) can be de need as a system that represents the characteristics and the actual situation of the organization. For instance, if we have a request to analyze a particular facet of the organization Embarked on an RTDW, the answer will be represented in the real state of the organization at the time of the request sending. Unlike most traditional data warehouses, an RTDW is seen to contain current data (real time) of the organization. Thus, the refreshing frequency plays a predominant role in RTDW. Generally, based on the literature a RTDW undergoes refreshments several times a day, which enables the decision makers to have access to the current data of the organization. To provide data freshness, performance, availability, prediction capabilities and to offer an integrated information repository to drive and tactical decisions, many approaches are proposed to treat the architectures RTDW and the data integration in the RTDW. As a part of this paper is organized as follows in many sections. In section 2, we introduce why RTDW is needed and explain the major comparison between the RTDW and the traditional DW. Section 3 presents some research studies that enable us to offer solutions to change the traditional Extract Transform Load (ETL) process to get closer to a RTDW. Section 4 describes some architecture proposed in some works. Section 5 presents some of the proposed approaches that address the problems of the data integration. Section 6 summarizes and discusses all the approaches studied in this paper. Section 7 summarizes the society impact of real time data. Finally, section 8 gives a conclusion and suggests some future research directions.

2. NEED FOR THE REAL TIME DATA WAREHOUSE

The data warehouse can be updated either conventionally or in real time. The drawback of the traditional method is that the data content is not updated, therefore, bad decisions may be made. In fact, there is a need for near real-time data warehouse where some data are updated in real time and the remaining data are traditionally refreshed. Thus, there will be an overload on the source system because only critical data will be frequently extracted from the source system or strategic decisions are made using old data. Storing data in near-real time will reduce the latency between business transactions to operational sources and their appearance in the data warehouse. This facilitates the analysis of more recent data and makes decision making faster. However, data warehouses and business intelligence real time applications have been proposed to answer exactly the types of questions that users would like to ask for real-time data. A Real-Time Data Warehouse enables data to store the data at a time when they are produced and immediately captured, cleaned and stored within the structure of the data warehouse. Then, traditional refreshment cycles are no longer valid. The data warehouse must be able to read the same data that move around the operating systems at the same time of its generation. The purpose of RTDW is to enable enterprises to rapidly access information and notify the user or decision-making system to react almost immediately to the information. There are two arguments that justify the use of real-time data warehouse. (1) The traditional data warehouse

which provides the state of the company at a specific time in the past (every day, every week or every month). However, some applications require a smaller temporal resolution (every hour). The real-time warehouse enables the organization to find variants of its state even within a day. (2) The second argument focuses on the fact that when a large volume of data is entered into the transaction systems in a single day (financial sector); processing ETL sometimes causes problems [1]. The operation of the real time warehouse suggests rather frequent and regular additions in a single day. Table1 presents a detailed comparison between the traditional data warehouse environment and the real time data warehouse environment. In the following sections, we will present some approaches that can move closer

Table 1. Major differences between a Traditional DW and an RTDW

Traditional Data Warehousing	Real Time Data Warehousing
Updating data Historical data periodically	Updating Real-time data
For strategic decisions only	For strategic and tactical decisions
Results hard to measure	Results measured with operations
Highly restrictive reporting used to check the pattern to make some prediction	Flexible ad hoc reporting and machine
Moderate user currency	Results measured with operations
Daily, weekly, monthly data concurrency is acceptable	Only data available within minutes is acceptable

Discussed RTDW and other approaches which represent an RTDW architecture and integration of real-time data. Closer to a Real time Data warehouse In most cases, updating a DW also means putting it out of use (shutdown) during the updating or at least this one makes its use much more di cult and with poor performances. Using a warehouse for an update may also cause some inconsistencies in the results which are returned by the query execution. They will be rarely logical and correct if interviewed data are updated at the same time. The criteria required for continuous updates without involving a shutdown are generally inconsistent with traditional ETL tools. To address this problem, new solutions specialize in ETL real-time and data loading. There are also solutions modifying conventional ETL systems in order to load a warehouse on a frequency approximating the real-time. Existing ETL system can be modified to perform real-time or near realtime data warehouse loading. Some of these solutions are described in [3]:

Near Real-Time ETL: This is first solution consists in eliminating the real-time reality of the choice if the need is absent. We could simply increase the frequency of loading new data into the warehouse. For example, a load that is usually done on a weekly basis could be executed once a day. This approach may, however, involve shutdown problems. This change would enable the users of the DW to have access to more recent data without having to make major modifications to the loading process or data model. Not being the real-time reality, the near-real time can be a good first low cost solution.

Direct Trickle: If an application requires real time, the simplest approach would be to continuously load the DW with the new data. In this way, we would eliminate any intermediate storage step. However, this solution can cause a loss of performance when the exploitation of the DW by one or more user(s) since the update of a warehouse itself can require a lot of work from the machine that hosts it.

Trickle and Flip:

This approach involves making the second partition of the fact table of the data warehouse on which we make the load. On a periodic basis, we replace the fact table of the warehouse with the second partition, which is responsible for new data. In this way, the searchable fact table is updated at a low frequency to limit the performance loss during its operation. It also contains all the new data since the last update. This solution can be used according to a cycle, which can vary from hours to minutes.

External real-time data cache: This approach consists in storing the data in real time in a real-time memory (Real Time Data Cache) external to the warehouse by completely avoiding any potential problem of performance issue. The Real Time Data Cache can be simply another dedicated warehouse loading of the storage and data processing. The applications which handle a big volume of data or which ask for a very short processing time could bane t from this solution. The major disadvantage of this solution is that it involves an additional database that must be installed and maintained. The work done to realize this approach is higher, but the costs that would be spent to buy more e cient equipment or additional memory are justified in many cases by the use of this method. The authors proposed some challenges and possible solutions for near real time ETL. They identified two problems for each extraction, transformation and loading phase. In the extraction phase, there are problems, such as the integration of multiple heterogeneous data sources solved using Change Data Capture with the stream processor and data integration tools. The second problem of this phase is the overload data source using a service of updating. In the transformation phase, the authors identified two problems, the first is master data overhead and the second focuses on the need immediate server to aggregate data. The solution for the first is to maintain a master cache of data and database queue. However, instead of transforming and loading the data, the solution to the second problem consists in switching both tasks, in other words, loading data first, and then transforming them, which could possibly reduce the time consumed for aggregation. During the loading phase, problems are: per-romance degradation and OLAP internal inconsistencies. Among the solutions who allow solving these problems are: the staging table (Trickle and Flip), amultiple stage trickle and ip, organizing outside the DW update period, snap-shot data, Real Time Data Cache and layer based view have been proposed. A part from this work not much work has been done on the investigation ETL challenges in near real time.

3. REAL TIME DATA WAREHOUSE ARCHITECTURE

Huge Work has already done on real time data warehouse architecture. As an example of lot of work done on RTDW proposed architecture, we will introduce architecture in the following approaches.

3.1 Architecture of Vassiliadis et al., 2009

The general proposed architecture called as a RTDW is detected in Fig.1. It deals with a continuous loading as opposed to the periodic basic in traditional approaches. It is composed of three main parts: (1) Data source hosting the data production systems that populate the data warehouse, (2) An intermediate

Data Processing Area (DPA) where the cleaning, extraction and transformation of the data takes place and (3) the data warehouse. Each data source hosts a Source Flow Regulator (SFlowR) module that is re-

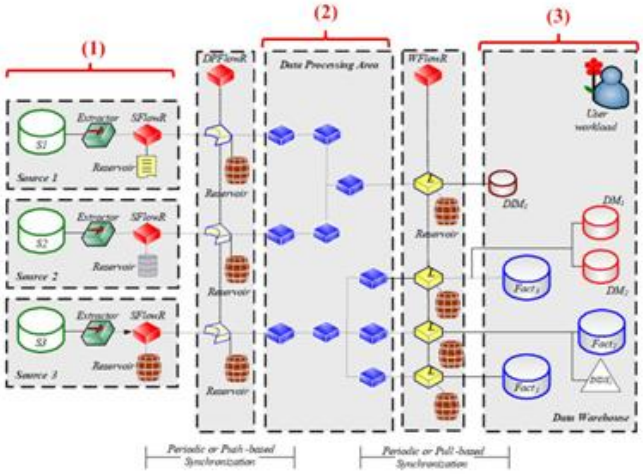

Figure.1. The Proposed Architecture of near real time data warehouse [9]

Responsible for the identification of relevant changes and propagates them towards the DW. Then, the Data Processing Flow Regulator (DPFlowR) module is responsible for deciding which source is ready to transmit the data. A Warehouse Flow Regulator (WFlowR) orchestrates the propagation of data from the DPA to the warehouse. This propagation is based on the current work load from end users posing queries and the requirements for data freshness, ETL throughput and query response time [9].

3.2 Architecture of Obali et al., 2013

The proposed RTDW assures the access to acquired data from different data sources in near real time [7]. The components of this architecture are Web Service Client, Web Service Provider, Metadata, ETL, Real Time Partition, Data Warehouse and Real Time Data Integration (Fig.2). Web Service (WS) Client is used to get the data changes (known as Change

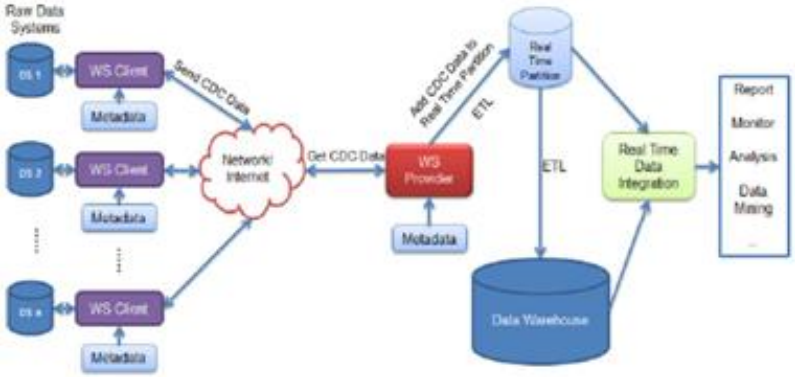

Figure. 2. Web Services Based Real Time Data Warehouse Architecture [7]

Data Capture - CDC) from raw data systems and send them to the Web Service Provider by calling the related web service. The data that is required to DW are collected from the operational systems and other external sources. Data capturing (CDC) techniques in use include: source data extraction, log capture, triggered capture, application-assisted capture, timestamp-based capture, and le comparison capture [7]. A Web Service Provider is basically a web service which gets data sent by RTDW Web Service Client and adds them to Real Time Partition. This component gets Data Transfer Object which is sent by RTDW Web Service Client, and decomposed them into two parts: data and metadata. The RTDW Web Service uses metadata to generate the Structured Query Language (SQL) via SQL-Generator to insert the data into RTDW log tables then execute this generated SQL on RTDW database and insert the data. In the Real Time Partition, the authors used three stages: the rst consists in putting the CDC data into a related warehouse log table. The second step consists in cleaning the CDC log data on demand and the last step is the aggregation of CDC data cleaned on demand. The component real time data integration is used to integrate the data both in Real Time Partition and Data Warehouse. When a user sends a query to this component; if query only wants historical data, then this component sends the query to Data Warehouse; if query wants both historical and instant data, then this component rewrites the query to get and integrate data [7]. Approaches that treat Real Time Data Integration to make data integration near real-time, different approaches [4],[2],[5] have treated the integration of data in real-time data warehouse.

4.1 Approach Lebdaoui et al., 2013

In [4], the authors handled the problems of data integrity issues towards the need for accessing to the RTDW and introduced an IA-RTDWg model that preserves, at the same time, the data integrity and availability. In this work, the authors discussed the integrity of data regarding the violation of RTDWg time constraints. They also presented their model which is based on a temporary duplication of fact tables that are concerned by the change of data, while keeping their integrity constraints. To manage the change in real-time data, RTDW uses specific criteria or measures as: real-time tables, real time partition and intelligent search tools to prioritize the integration of data according to their importance. The RTDW systems must ensure data integrity. If someone has access to the data just captured by the CDC mechanism or transformed by ETL tools and modifies illegally them, therefore integrity is stripped. If a

program or a tool in a DW solution has not processed the data according to predefined rules, and if DW step is ignored or skipped, integrity can also be corrupted.

4.2 Approach Ferreira et al., 2013

On the basis of the industry and theoretical ideas, [2] defined a data warehouse architecture for the constant data integration without compromising the performances of the requests, and assessed its ability to provide real-time. In this real-time data warehouse, the authors defined a dynamic storage component and a static storage component to represent the recently data integrated and the rest of the data, respectively, with appropriate choices regarding how the components merge. They also proposed design choices concerning the calculation of the query mechanisms and estimated the alternatives to conclude what is the most effective implementation of these mechanisms. The authors proposed the RTDW architecture from a basic DW architecture. This approach enables to updates the continuous data while keeping the most recent information in DW and enabling any desired degree of freshness to queries. The architecture is made of three main components. These are Dynamic Data Warehouse (DW-D), the Static Data Warehouse (S-DW), and fusion. In [2], the authors also proposed a solution that allows the integration of new data in real time. The solution involves effective loading data into a component that holds the most recent information and offers fast integration mechanisms, D-DW. This component was created to allow a very fast rate of integration. Queries run quickly and introduce the lower efficiency charging, due to small amounts of data. This D-DW component can run in memory. However, a data warehouse is also an extensive set of historical data, and the processing of such data is a task with a high cost. These historical data are stored in the S-DW component and, given that the D-DW component operates independently of S-DW. With this architecture (Fig.3.), it is possible to provide guarantees of constant integration, new data and, at the same time, access to both the most and the least, without significant additional performance cost.

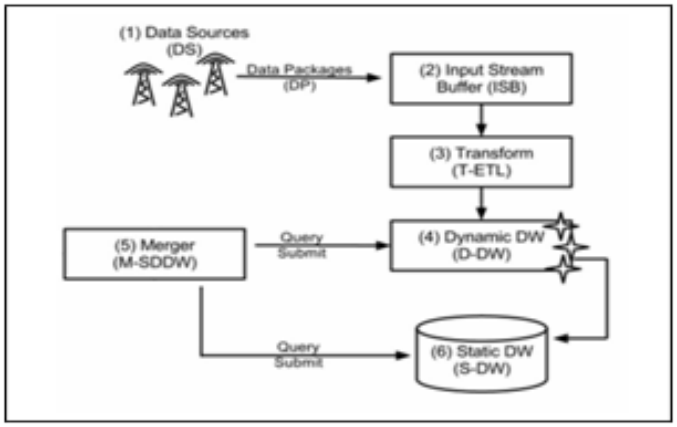

Figure.3. Sketch of proposed RTDW [2]

4.3 Approach Lebdaoui et al., 2014

For quick and effective decisions for the future, it is necessary that the data warehouse reflects the real changes in operational data and provides the freshest data for analysis systems. In [5], the authors presented a new method of improving data integration, while preserving the data integrity respecting the existing safety rules. They treated the problem of the integration of large data in the data warehouse in a short time and proposed a new model called Divide-Join Data Integration (DJ-DI). This model, which is built on dividing the volume of data changes and the duplication of the fact table, shows a real improvement in data integration rate compared with the normal configuration. The authors conducted various simulations DJ-DI model as an experimental platform. The results show that the DJ-DI model offers a remarkable improvement of data integration rate.

5. DISCUSSION

It should be noted that, the majority of the presented studies of a RTDW did not specify the ETL real time architecture. Only the work of Wibowo A., [10] is specialized in the identification of the problems and their solutions at the level of every phase in the ETL process. The works [4],[2] and [5] treat the integration data in the RTDW. Furthermore, all the presented approaches are concentrated

Table.2. Summary of the literature review

Paper	Proposed Approach	X1	X2	X3	X4	X5	X6
Obali N. et al., 2013	Construction a RTDW based on the Web services	Yes	Not treated	Yes	Change Data Capture Log Table	No	No
Lebdaoui I., et al., 2013	Presentation a new model IA-RTDWg to protect at the integrity and the accessibility of the data at the same time	Yes	Not treated	No	Change Data Capture Temporary Table Real Time partition	Yes	No
Lebdaoui I., et al., 2014	Proposal a new model called DJ-DI to handle the problem of the integration big data in the data warehouse in a short time	Yes	Not treated	Yes	–	Yes	No
Wibowo A., 2015	Identification of the problems and their solutions at the level of every phase of the ETL process	No	Near Real Time	No	–	No	No
Ferreira N., et al., 2013	Proposal a solution for the real time storing, with the capacity of integration of constant data and / or at high speed	Yes	Not treated	Yes	Change Data Capture	Yes	No

On the architecture of the RTDW and not detailed the design and the implementation of these architectures. Table 2 gives a summary of the literature review which is based on some criteria: X1 (RTDW Architecture), X2 (ETL Real Time), X3 (Prototype), X4 (The techniques used in the approach), X5 (Data Integra-tion), X6 (Modeling approach). There is no work functionality, recurring to a simulation using the TPC-H benchmark, performing continuous data integration at various time rates against the execution of various simultaneous query workloads, for data warehouses with different scale sizes. And we could achieve real-time data performance in exchange for an average increase of query execution time to load data in RTDW This should be considered the price to pay for real-time capability within the data warehouse. We could develop an Real Time Data Loader tool [RTDL] which will integrate the methodology with extraction and transformation routines for the OLTP systems. There is also room for optimizing the query instructions used for our methods.

6. SOCIETY IMPACT OF REAL TIMA DATA

Real time data use up-to-date and consistent data in analytical and transactional systems. Achieve a comprehensive view of business using all data sources including log files, databases, sensors, and messaging systems. Accelerate building streaming data pipelines using prebuilt integration and wizards-based development. Enable timely insight for better operational decision making. Stream combines non-intrusive, real-time change data capture capabilities with in-flight data processing to deliver timely and enriched data to the rest of the enterprise. It is an end-to-end, enterprise-grade platform with built-in stream analytics and data visualization, and delivers real-time insights while moving the data with sub-second latency. Stream provides an intuitive development experience with wizard-based user interface and speeds time-to-deployment with pre-built data pipelines. Using an SQL-like language, it is familiar to both business analysts and developers. With Stream, you adopt future-proof, smart data architecture for accelerated innovation. Real-time analytics is the ability of a business enterprise to use all available enterprise data when needed. A crucial feature of real-time analytics is that the available systems and setup should be able to quickly generate analytics based on the data received, ideally within a minute of the data being generated. A big advantage of real-time analytics is the freshness and the context of the data. Organizations can reap a lot of benefits by accessing real-time analytics purely because of their close relevance to market realities

7. CONCLUSION

In this paper, we went through about tradition and real time data warehouse by indicating the difference approaches of real time data integration of the problems and the solutions to move closer to this RTDW. We did survey to review literature review of existing data warehouse architecture, and introduced the changes in the ETL process to deal with real time data warehouse systems, and treats the integration data in the RTDW, and described the real time data impact on society and benefits. As future work we intend to develop an tool as Real Time Data Loader (RTDL) which will integrate this methodology with extraction and transformation routines for the OLTP systems. There is also room for optimizing the query instructions used for our methods. We will think about a data loading methods to load the data in to RTDW and real-time ETL process taking into account the characteristics of data collected in real time.

8. REFERENCES

[1].Real Time Data Streaming Tools and Technologies – An Overview.

[2].https://www.algoworks.com/blog/real-time-data-streaming-tools-and-technologies/ Year of Publication: July 2017.

[3].Isaac Sacolick. Real-time data processing with data streaming new tools for a new era. Year of Publication: 2018.

[4].Babak Yadranjiaghdam ; Nathan Pool ; Nasseh Tabrizi. A Survey on Real-Time Big Data Analytics. International Conference on Computational Science and Computational Intelligence (CSCI). Year of Publication: 2016

[5].Michael H. Brackett. Streaming Analytics Captures Real-Time Intelligence. Wolfram Wingerath, Felix Gessert Baqend GmbH, Erik Witt Baqend GmbH.

[6].Real-Time Data Management for Big Data Year of Publication: 2017.

[7]. Felix Gessert and Norbert Ritte. Scalable Data Management: NoSQL Data Stores in Research and Practice. In 32nd IEEE International Conference on Data Engineering, ICDE 2016

[8]. FelixGessert, MichaelSchaarschmidt, Wolfram Wingerath, Erik Witt, Eiko Yoneki, and Norbert Ritter. Quaestor: Query Web Caching for Database-as–a-Service Providers. Proceedings of the 43rd International Conference on Very Large Data Bases (2017), Year of Publication: 2017.

[9].Gessert, Wolfram Wingerath, Steen Friedrich, and Norbert Ritter. Year of Publication: 2016.

[10].NoSQL Database Systems: A Survey and Decision Guidance. Computer Science - Research and Development Year of Published 2017.

[11].Felix Gessert, Wolfram Wingerath, and Norbert Ritter. Year of Publication: 2017.

[12].Real-Time Databases Explained: Why Meteor, Rethink DB, Parse and Firebase Don't Scale. [13].Baqend Tech Blog Year of Published 2017.

[14]. R.G. Little, M.L.Gibson, Perceived influences on implementing data warehousing" IEEE Transactions on software engineering, Volume 29, Issue: four, page: 290 - 296, Digital Object Identifier: 10.1109/TSE.2003.1191794, Year of Publication: 2003.