Problems and Available Solutions On The Stage of Extract, Transform, and Loading In Near Real-Time Data Warehousing (A Literature Study)

Ardianto Wibowo
Department of Informatics Engineering
Politeknik Caltex Riau
Pekanbaru, Indonesia.
email: ardie@pcr.ac.id

Abstract – In the traditional ETL (Extract Transform Loading), refreshment of data warehouse must be done in off peak hours. It means that all operational and analysis stopped from their all activities. It cause the level of freshness of data in the data warehouse is not indicating the latest operational transaction. This problem is called by data latency. Near real time data warehousing is used to be a solution for this problem. It update data warehouse in near real time manner, immediately after change data detected in data source. Thus, data latency can be minimized. In development, near real time data warehousing have problems where previously not found on the traditional ETL. This paper aims to convey the problems and available solutions at each stage in the near real time data warehousing, i.e. extraction, transformation, and loading. The problems and available solutions are based on literature review from other research that focusing on near real time data warehousing problem.

Keywords: ETL; Near Real Time Data Warehousing; Data Latency; Data Warehouse

1. INTRODUCTION

Data warehouse is updated through ETL (Extract, Transform, and Loading) process. ETL have responsibility to detect relevant change data, extract it into staging area, transform it into appropriate format, and load it into data warehouse table [1].

Traditionally, ETL is updating data warehouse periodically [2]. This implies that data warehouse is not relevant to the current condition [3], where there is real time data between two updating process. Thus, it makes less accurate analysis result.

The other problem is that traditional ETL should be done at off peak hours [2]. It means operational and analysis activity must stop off all activities [1,3,4,5]. It make serious problem for a system that must be running 7x24 hours [6].

Based on these problems, there should be a mechanism to updating data warehouse immediately after change data detected so that the user need to the latest data can be met. This mechanism is called by near real time data warehousing [1,7]. Other terms is operational data warehouse [8] or real time ETL. On the near real time data warehousing, loading data process into data warehouse is performed continuously. It is different with traditional approach that works periodically [9].

Near real time data warehousing have problems previously not found on the traditional ETL. From related research that focusing on near real time data warehousing problems [3,10], not obvious how the portion of the failure occurred at extraction, transformation, or loading stages. This paper aims to propose a taxonomy that contains the problems and available solutions at these each stage. These problems and available solutions are based on literature review from other research that focusing on near real time data warehousing problem. With this taxonomy, is expected that make the further research easier to get the problem focus and the contribution that will be developed.

2. RELATED WORK

There are few studies that have grouped the problems in the near real-time data warehousing. The problems can be divided into 5 challenges, namely enabling real time ETL, modeling real time fact tables, OLAP query versus changing data, scalability and query contention, and real time alerting [3]. The other research divide it into 3 groups, namely enabling real time ETL, enabling real time BI, and data stream management system [10]

3. PROBLEM AND AVAILABLE SOLUTIONS IN NEAR REAL TIME DATA WAREHOUSING

A. Extraction stage

Extraction is the process of getting data from data source. There are two problems in extraction stage as follows:

a. Multiple – heterogeneous data source integration

Data source can be divided into two parts, namely stored data set and data stream. Stored data set is data that can be used over and over again, and infrequent updating process. Data stream is data that the use not repeatedly and constantly changing [11]. Example of data stream is in finance application, network traffic monitoring, click stream web application, sensor, email, and telephone call data detail [12].

To handle data stream, stream processor can be used [11]. Whereas to handling stored data set, CDC – Change Data Capture is used [11,13]. To integrate both of data, stream processor and CDC are connected to a message queue that connected in a data integration tool [11].

b. Data source overload

Reading data source continuously make overload that disturb operational activities. Due in addition to serve the operational transactions, the data source must also serve a reading by CDC.

To solve this, more efficient extraction method in CDC has developed, which is called by update significance and number of record changed method. This method aims to obtain priority data to be carried on each extraction. Change data that don't fit the priority category will be extracted using traditional ETL [7]. This method is shown at Figure 1. Another way is create a special format for log table in CDC-trigger [14].

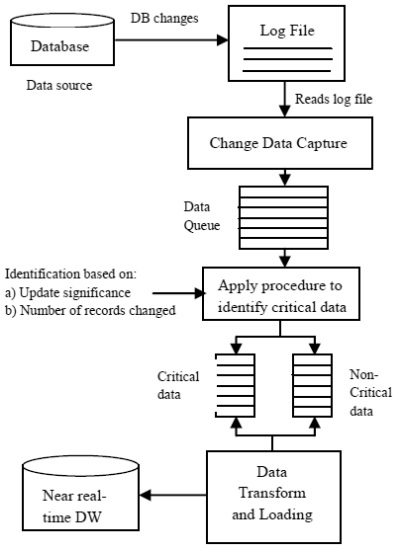

Fig.1. Update significance and number of record changed method

B. Transformation stage

Transformation is a process to adjust the data obtained from the data source into a predetermined format. There are two problems in transformation stage as follows:

a. Master data overhead

Data stored in data warehouse can be divided into two parts, namely *master data* and *transactional data* [4]. Master data is data that is not frequently changed. For example is product or customer. In the data warehouse, master data is implemented by dimension table. Transaction data is frequently changing data according with transaction occurred in data source. For example is sales transaction. In data warehouse, transaction data is implemented by fact table.

Every data warehouse refreshment process is based on transaction data generated. However, this process also need master data. Master data will be used for transaction data join process. Thus, same master data will be frequently extracted. It problem is called by master data overhead.

To solve this, master data is placed on a cache, while real time data is placed in the database queue. Furthermore, the join is done between real time data on the database queue with master data on cache for each transaction. This mechanism is illustrated at Figure 2 as follows [4]:

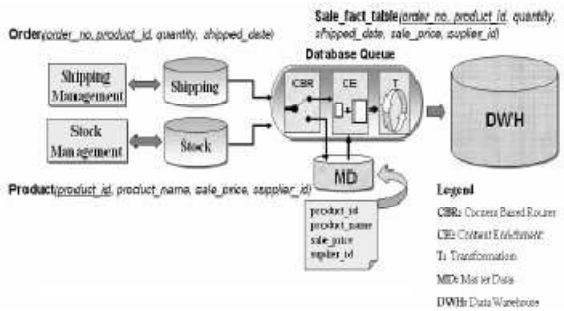

Fig.2. Mechanism of handle master data overhead

b. Require intermediate server to perform data aggregation

Transformation process is done before data is loaded into data warehouse. In the traditional ETL, transformation processes a group of data in the staging area with ETL tools. On the near real time data warehousing, each data warehouse refreshment process only carry one or several small amounts of data. This resulted that the transformation process can't be performed on each data warehouse refreshment cycle. To solve this, a method with the name ELT (Extract Load Transform) can be used. With ELT, transformation process is executed in the data warehouse [8,15] as shown at Figure 3 [15].

According to Figure 3, extracted data from data source will be loaded directly into data warehouse. It resulting data warehouse contains unclean copy of data source. Therefore, transformation process to reshape data into appropriate format is needed.

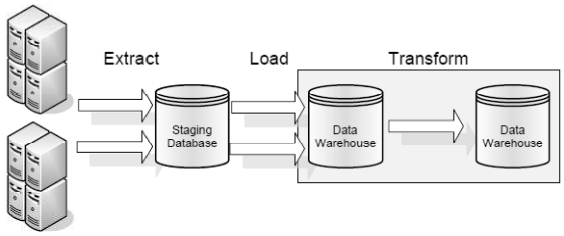

Fig.3. ELT method

C. *Loading stage*

Loading is a final stage from ETL process, where clean data from transformation process is loaded into appropriate data warehouse table. The problem occurred when there are transactions during OLAP analysis. Transaction data will overlap with OLAP process [6]. As a result, there will be performance degradation in analysis process. Another consequence is the occurrence of OLAP internal inconsistency.

a. Performance degradation

To minimize the performance degradation in analysis process, staging tables (temporary tables) can be used as a solution. Staging tables is table which has exactly same format with data warehouse destination tables. This staging tables will be used to receives and stores real time data temporarily.

Data warehouse destination table will be updated from this staging table periodically. This solution is named by trickle and flip [3]. This solution was developed further into *multi-stage trickle and flip* as shown at Figure 4 [2].

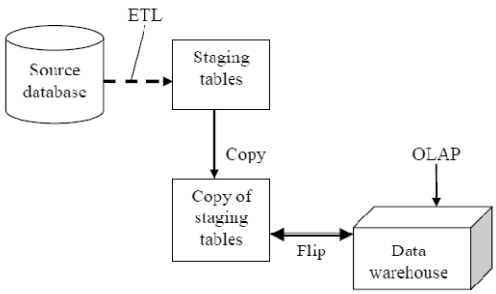

Fig.4. Multi-stage trickle and flip

b. OLAP internal inconsistency

OLAP is designed to operate with static data. There is no mechanism to prevent data modification to data being used by an OLAP process. If modifications of data occur at the same time with an OLAP activity that uses that data, OLAP will issue inconsistent result. This problem is named by OLAP internal inconsistency [3]. This problem can occur in OLAP operation such as roll up – drill down. Inconsistencies will occur between aggregation and detail result.

To prevent this, analysis proses on staging table solution can be done outside the refreshment period of data warehouse table [3]. Other solution is taking a snapshot of data from the data warehouse table and the use of RTDC (Real Time Data Cache). The snapshot data is used for analysis process, whereas data warehouse table is used to save real time data permanently. RTDC is the usage of external cache between data source and data warehouse. This cache will be used to temporarily store real time data. Cache will be read periodically to move its content to the data warehouse table. If necessary, JIT (Just in Time) method can be used to combine data in the data warehouse table and cache in an analysis process [3].

Another solution is combining row lock with layer - view that generated dynamically. This method is named by layer-based view as shown at Figure 5 [16].

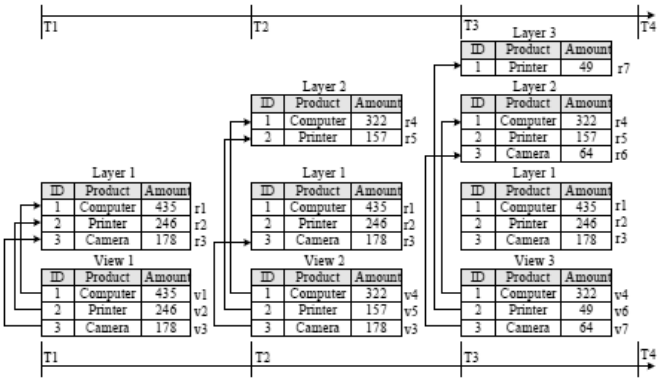

Fig.5. Layer-based view

According to Figure 5, when rows in a table are used for analysis process, these rows will have row lock. If at the same time these rows be a purpose for a data source transaction, new layer will be generated automatically to store these change data. For each analysis process, a view which involves generated layer will be created. The aim from this view is to provide the latest data for each analysis process.

D. *Problem and available solutions taxonomy results*

Based on the explanation above, the problems and available solutions that have been developed in the process of extraction, transformation, and loading on near real-time data warehouse can be summed up into a taxonomy as shown in Figure 6 as follows:

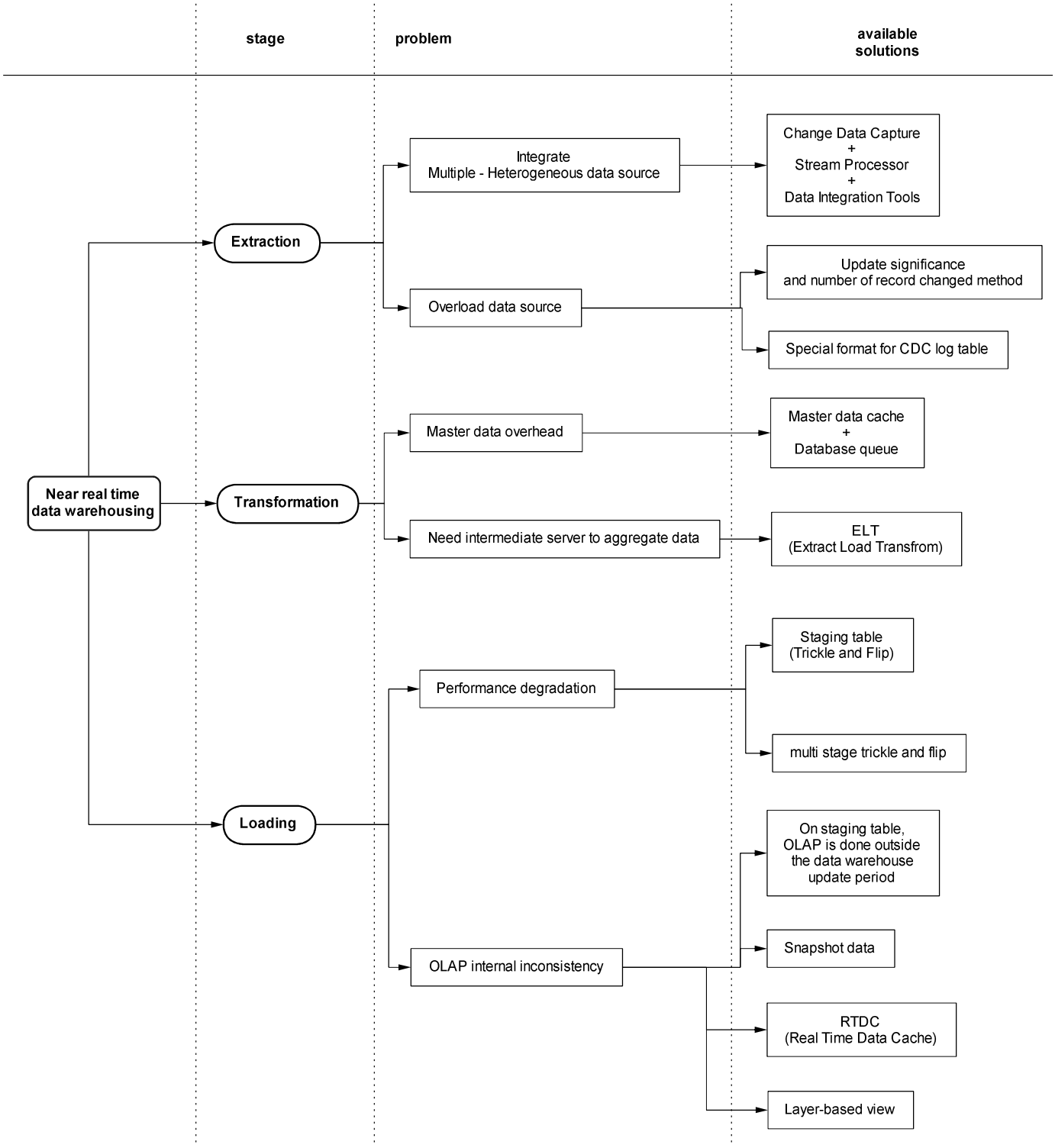

Fig.6. ETL problem and available solutions in Near Real Time Data Warehouse taxonomy result

4. CONCLUSION

This paper has propose a taxonomy contains the problems and available solutions on the stages of extraction, transformation, and loading in the near real time data warehousing based on literature review. With this result, further research associated with near real-time data warehousing can use it to get the focus issues and their contribution that will be given.

REFERENCES

[1] R. J. Santos and J. Bernardino, "Real-Time Data Warehouse Loading Methodology," *IDEAS '08 Proceedings of the 2008 international symposium on Database engineering & applications* , pp. 49-58, 2008.

[2] J. Zuters, "Near Real-time Data Warehousing with Multi-stage Trickle & Flip," in *10th International Conference, BIR 2011*, Riga, 2011, pp. 73-82.

[3] J. Langseth. (2004, Aug.) dssresources.com. [Online]. http://dssresources.com/papers/features/langseth/langseth02082004.html

[4] A. M. Naeem, G. Dobbie, and G. Weber, "An Event-Based Near Real-Time Data Integration Architecture," in *EDOCW '08 Proceedings of the 2008 12th Enterprise Distributed Object Computing Conference Workshops* , Washington, 2008, pp. 401-404.

[5] J. Guerra and D. A. Andrews, "Creating a Real Time Data Warehouse," 2011.

[6] D. Agrawal, "The Reality of Real-Time Business Intelligence," in *Business Intelligence for the Real-Time Enterprise*. Auckland: Springer Berlin Heidelberg, 2009, pp. 75-88.

[7] T. Jain, R. S, and S. Saluja, "Refreshing Datawarehouse in Near Real-Time," *International Journal of Computer Applications*, vol. 46, 2012.

[8] G. Swetha, D. Karuranithi, and K. A. Laksmi, "Data Integration Models for Operational Data Warehousing," *International Journal of Computer Science and Mobile Computing*, 2014.

[9] K. Kakish and T. A. Kraft, "ETL Evolution for Real-Time Data Warehousing," in *Proceedings of the Conference on Information System Applied Research*, New Orleans, 2012.

[10] R. S, S. B. B, and N. K. Karthikeyan, "From Data Warehouse to Streaming Data Warehouse : A Survey on the Challenges for Real Time Data Warehousing and Available Solutions," *International Journal of Computer Applications*, 2013.

[11] M. N. Tho and A. M. Tjoa, "Zero-Latency Data Warehousing for Heterogeneous Data Sources and Continuous Data Streams," in *5th International Conference on Information Integration, Web Application, and Services*, Jakarta, 2004.

[12] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, "Models and Issues in Data Stream Systems".

[13] Attunity, "Efficient and Real Time Data Integration With Change Data Capture," 2009.

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

[15] R. J. Davenport, "ETL vs ELT A Subjective View," Data Academy, 2008.

[16] Z. Lin, Y. Lai, C. Lin, Y. Xie, and Q. You, "Maintaining Internal Consistency of Report for Real Time OLAP with Layer Based View," *Springer*, 2011.