A Big Data Perspective of Current ETL Techniques

K V Phanikanth
Dept. of Information Technology
Alliance College of Engineering and Design
Bangalore, India
phani.kv83@gmail.com

Sithu D. Sudarsan
Group Manager
ABB Corporate Research
Bangalore, India
sdsudarsan@gmail.com

***Abstract*— Dynamic data stream processing using real time ETL techniques is currently a high concern as the amount of data generated is increasing day by day with the emergence of Internet of Things, Big Data and Cloud. Data streams are characterized by huge volume that can arrive with a high velocity and in different formats from multiple sources. Therefore, real time ETL techniques should be capable of processing the data to extract value out of it by addressing the issues related to these characteristics that are associated with data streams. In this work, we asses and analyze the capability of existing ETL techniques to handle dynamic data streams and we present whether the existing techniques are relevant in the present situation.**

Keywords — ETL; data warehouse; data streams; continuous query; big data;

I. INTRODUCTION

Data Management in the electronic and computing era dawned with data model and design using techniques like normalization. The very idea was based on the fact that information in its original form is mostly unstructured and contains redundant as well as information that can be derived from primitives. Of the several reasons, storage cost, communication cost and computing cost were of key. Given the fact that each MB of storage costed close to USD700 in early 80s, about USD1200/Mbps around 1998 and about USD10000/MIPS in the 80s, these considerations made real economic sense. Yet another aspect related to data was about quality related to missing data, outliers and inconsistent formats. All these issues were addressed by developing techniques to extract, transform and load (ETL) data in to specific databases. However, time is not static and technology has progressed. As of today, the cost of storage, computing or communication are negligible. The amount of data generated has also increased tremendously making traditional data processing and data warehousing techniques becoming infeasible. This has resulted in emergence of Big Data paradigm. Increasing demands of Big Data processing has led to many challenges that arise due to its characteristics such as volume, velocity, variety, value and veracity. Volume refers to the huge amount of data that needs to be processed. Velocity refers to the speed at which the data arrives for processing. Variety refers to the data that can arrive from different sources with different formats. Value refers to the valuable information that can be extracted out of the raw data to make important business decisions. Veracity refers to the authenticity of the data. The biggest challenge in processing big data that arrives in streams is to extract the valuable information out of it by addressing the issues related to its volume, velocity, variety and veracity.

In order to ease the process of querying the useful information, the processed data are stored in the Data warehouse. The data warehouses are often designed to support analytical data processing. Usually the ETL process forms an inseparable part of data warehouse which is used for processing the data. The evolution of ETL process can be divided into three following categories:

- **Traditional ETL** that can handle processing of highly structured data. Usually databases are considered as source of input and it operates usually in batch processing mode.
- **Near-real time ETL** that process the structured data more efficiently than traditional ETL by reducing the refresh time and thus attempts to maintain the freshness of processed data in data warehouse.
- **Real-time ETL** which processes the data streams in real time that can arrive from different sources in various formats and characterized by huge volume and high velocity.

Many of the existing ETL techniques process the static data obtained usually from databases that are highly structured and populate the data warehouse in the required format for further decision making. Some of the ETL techniques attempt to handle processing of data streams, but data streams are characterized by volume, velocity [1][2] and variety [3] which needs to be addressed for processing data. From this perspective, the currently available so called real time ETL approaches are inadequate to address all these characteristics of data streams. Therefore there is a need to redefine the real time ETL process that can address the issues related to processing of data streams.

The rest of the paper is organized as follows: Section 2 presents related work in this area. Section 3 discusses the

978-1-5090-2576-3/16/$31.00 ©2016 IEEE

available real time ETL tools and their drawbacks. Section 4 analyses the capability of current ETL techniques to handle the maximum amount of data and section 5 focuses on the conclusion and future work.

II. RELATED WORK

A. *Extracting useful data from data streams by applying a continuous query onto the arriving data*

The different sources are assumed to produce data streams but not capable of storing the data by themselves. Therefore there is a need to extract the data in real time and process it, so that the users can query the processed data to make appropriate decisions. In case if the arriving data is not captured properly the data would be lost permanently. As we know the data streams are characterized by volume and velocity which has to be addressed to capture the data. Therefore the existing methodologies use continuous query techniques [4] to extract only the data that are useful with respect to a particular context. The rest of the original data are discarded.

B. *Transforming the extracted data into the form required by the data warehouse*

Many of the existing techniques address transforming of highly structured data into fact tables and dimension tables as required by the traditional data warehouse techniques. But the data streams are characterized by heterogeneity as well. Therefore there is a need to address the challenges that arise due to the heterogeneity of the data. Some of the existing techniques attempt to address the heterogeneity of the data by creating a semantic model by constructing ontologies and linked RDFs.

C. *Integration of real time and traditional data warehouse architectures*

Many of the existing techniques attempts to enhance the traditional data warehouse architecture to handle real time data by adding a real time data storage component to the existing architecture. The data streams after being processed are loaded into the real time data storage component from which the users can query for the fresh data and after some time duration the data from the real time data storage area can be moved to the historical storage area for later use. This approach can be observed in fig. 1 which shows the complete ETL framework.

D. *Synchronization of data updates and querying the data warehouse*

Continuous loading and updating of processed data into the data warehouse is an essential process when handling data streams. But this affects the performance of fetching the query results from the data warehouse. This leads to the problem of query contention. Therefore there is a need to synchronize the process of data updation and query operations. Many of the existing techniques attempts to address this issue by introducing buffering techniques to capture the processed real time data and updating the data warehouse once for all when the triggering conditions are satisfied.

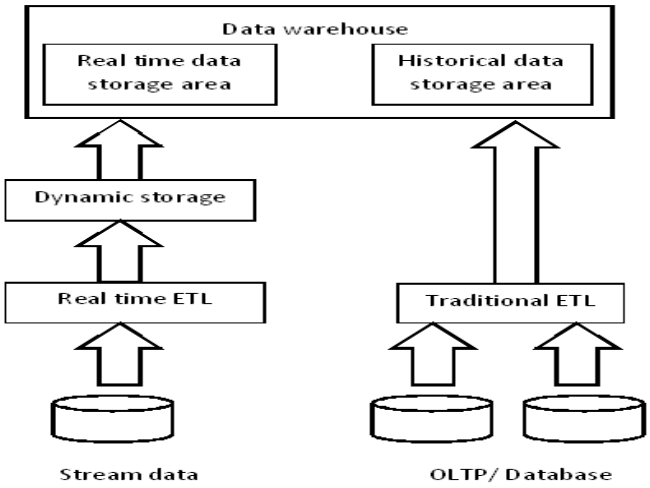

Fig. 1 ETL Framework

III. AVAILABLE ETL TECHNIQUES TO HANDLE DATA STREAMS

Authors of [5] propose a methodology to handle data streams using real-time ETL. The proposed methodology separates out real-time ETL from historical ETL and introduces a dynamic storage area to address the problem of batch updates by synchronizing the data aggregation operation with real time data and to improve the freshness of the query results. The architecture does not specifically address the problems of handling unstructured data with huge volume and velocity.

Authors of [6], [7], [8] and [9] incorporate semantic technology into the ETL process to address the semantic heterogeneity of the data. The methodology in [6] creates a semantic data model by mapping the data sets onto the ontologies and loading the RDFs to the data warehouse. The methodology addresses the problem of integrating heterogeneous source data into a standard format and thus addressing one of the v's of big data characteristics namely variety and provides a way to organize the data through the RDF links which actually increases the accuracy of the results as well. This methodology does not consider volume and velocity of the data, as it is difficult to construct ontologies manually. The Authors of [9] propose a technique with semantic approach that makes use of linked RDFs to represent the data sets. Semantic filtering is applied for integrating large volume of heterogeneous streams on the fly. Semantic data streams are then filtered by applying continuous queries. The approach also makes use of

summarization techniques to discard some data when it crosses a limit that can be handled. Therefore the semantics of the original data is being lost if it arrives with huge volume and high velocity.

Authors of paper [10] propose a technique to handle data streams which includes a data stream processor and an operational data store (ODS). The proposed methodology applies continuous query on to the arriving data streams so that the irrelevant data can be filtered out and the data stream reduces to manageable size. Even after reducing the data stream, if it crosses a specified threshold level which can be stored on the memory, data are divided into equal parts and small samples are collected using sampling techniques and rest of the data are discarded. The proposed methodology tries to address velocity of the stream data up to some extent but if volume of the data increases then the architecture fails to process the data without losing some of the source data. The methodology also addresses the problem of synchronization of arriving data with data processing up to some extent but at the cost of losing the semantics of the original data.

Authors of [11] address the problem of synchronizing data aggregation operation and querying the real time data warehouse by proposing an algorithm called Integration Based Scheduling Approach (IBSA). IBSA can be divided into two parts. The first part details about triggering the ETL process when the data sets are arriving from different sources and the second part of the algorithm decides upon whether to invoke a thread that performs update of the data warehouse or to invoke a thread that queries the real time Data Warehouse. This technique addresses the problem of fair resource allocation for both updating and query operations. The proposed methodology attempts to address the challenges related to real time data warehouse to handle the data. The data streams can be characterized with volume, velocity and variety. The proposed technique fails to explain how the scheduling policy scales in the situation when the input data streams for updating the data warehouse is characterized by high volume and velocity and also when the query queue is full. It does not consider the heterogeneity of the data during the integration process as well.

Authors of [12] propose architecture to support both real time data warehouse to handle data streams and historical data warehouse to handle historical data. The methodology captures the continuous data by implementing multilevel caching technique which actually differentiates the freshness of the arriving data. The proposed methodology also introduces double mirror partitioning technique to synchronize the data warehouse update process and querying operations. As the proposed methodology makes use of the caching technique, the buffer should be of limited size. If the buffer size is increased, the efficiency reduces because of the reduced hit rate. Therefore when the data stream arrives with huge volume and velocity, some amount of data must be dropped as it contains limited buffer space. As a result of this, the semantics of the original data is lost.

Authors of [13] and [14] provide detailed information about the evolution of ETL process starting from traditional ETL to real time ETL to handle data streams. Different existing real time and near real time ETL techniques are discussed. The data streams that are considered for real time ETL process can also be characterized by volume and velocity and therefore there is a need to address these two characteristics in real time data processing. But the different techniques consolidated in these two works do not address any of the characteristics of data streams. The architecture provided for real time data processing makes use of a stream analysis engine during extraction process which actually looks for some specified patterns from the arriving data streams and rest of the data are discarded. As a result of this the semantics of the original data is lost.

The methodology proposed by [15], Provides a review on existing problems and available solutions of processing historical data as well as data streams. It also provides some views in addressing the problem of joining the historical data and real time data streams by using a buffering technique. While processing the data streams, the transformation phase cannot cope up with processing the continuously arriving fast data streams and therefore this paper discuss about a technique called ELT in which the transformation process on data is being done after loading the data on to the data warehouse. But the paper considers highly structured data and does not address the heterogeneity of the data. The methodology fails to address the problem of scaling with the volume and velocity associated with the data streams.

The architecture proposed in [16] implements a real time ETL engine which is responsible for processing the data streams and loading it on to the real time data warehouse. The real time ETL architecture consists of RBFs (Remote Buffer Framework) which is responsible for receiving the data streams from different sources. These RBFs are connected to RIFs (Remote Integrator Framework), whose function is to accumulate data from different RBFs and pass it on to real time ETL. The architecture clearly specifies that different sources can be connected to a single RBF. The data streams arriving from different sources can be characterized by volume and velocity. Therefore the RBFs when receiving arriving data streams from different sources may run out of memory space if the arriving data has huge volume and velocity in which case some amount of original data must be discarded, as a result of which the semantics of the original data is lost before even processing the data.

The methodology in [17] and [18] attempts to enhance traditional data warehouse to support real time data stream processing. The proposed methodology assumes that the same old traditional ETL architecture holds good for processing data streams with very few modifications to it. As the data streams arrives in real time, loading and refreshing time interval will be very less which reduces the data warehouse loading process significantly. Therefore the proposed methodology incorporates a modification, to isolate dynamic data warehouse component which supports data stream processing from traditional data warehouse. But, as data streams are always characterized by volume, velocity and variety, the traditional ETL component cannot cope up with these characteristics. The traditional ETL uses staging operation for processing the data which uses small sized fixed memory which cannot store huge volume of arriving data streams. The proposed architecture does not address the issues with velocity and variety of the data as well.

Authors of [19] propose a methodology to address real time data integration in the transformation phase of ETL process. This approach implements a technique called Divide Join - Data Integration (DJ- DI) whose behavior changes according to the size of arriving data. When a change in the operational data source tables are identified they are partitioned in to manageable size and join operation is performed on each partition. The approach considers big data as the source data and address only structured data but big data can be characterized by huge volume of variety of data arriving with high velocity.

The methodology in [20] implements web services based real time data warehouse architecture. The proposed methodology attempts to address real time data warehouse challenges by isolating real time data warehouse component from traditional data warehouse. As the architecture is based on web services for transferring the data from source to real time data warehouse component, it cannot support heterogeneous data types. As data streams are characterized by volume, velocity and variety, the real time data warehouse architecture must address the challenges that arise from these characteristics. The proposed architecture uses web services and does not address the challenges that arise with respect to network related issues such as traffic, bandwidth and so on because of volume and velocity of the arriving data streams.

The methodology in [21] attempts to address the challenges related to handling real time data warehouse by considering three aspects. They are, real time data extraction methods, maintaining consistency during integration and continuous loading of processed data. The real time data extraction method specified by the Authors, uses log analysis which is not suitable for data streams as the data streams are characterized by volume and velocity and because of which it becomes difficult to handle logs. The proposed methodology details about integrating the data from different sources but does not take into account the heterogeneity of the data during integration process. The methodology attempts to address the challenges that arise due to the volume of data by proving filtering functionality to remove less important data with respect to the context and as a result of this the semantics of the original data is lost.

The paper [22] attempts to address the feasibility of integrating the real time data and trade-offs between quality and availability of the data in data warehouse for querying in real time. The proposed methodology uses same old traditional ETL approach for handling the data streams which is not feasible as the traditional ETL cannot cope up with the challenges posed by volume and velocity of the data streams. The methodology does not address the challenges related to heterogeneity of the data streams as well and considers only the structured data.

IV. ANALYSIS

In this section, we analyze the current ETL techniques and tabulate the approaches to show the maximum amount of data that it can handle and we also categorize the approaches based on whether it can handle data in batches, near real time or real time.

TABLE I
Quantum of data considered by current ETL techniques

	Batch	Near- Real time	Real time
KB	--	--	--
MB	Xiaofang Li [5] J. Song [11]	Shao YiChuan [12]	--
GB	Srividya K. [6] J.Villanueva Chávez [7] L. Jiang [8] Imane Lebdaoui [19] L. Imane [22]	Marie-Aude Aufaure [9] F. Majeed [10] Kakish Kamal [13] Revathy S [14] A. Wibowo [15] Marcin Gorawski [16] Alfredo Cuzzocrea [17] Alfredo Cuzzocrea [18] M. Obalı [20] Rui Jia [21]	--
TB	--	--	--

The TABLE I show the quantum of data that the current ETL techniques are able to handle, while processing the data streams. Since none of the approaches are supporting beyond tera bytes of data, we have considered the maximum limit of data in terms of TBs. The Authors of current ETL techniques have put lot of efforts in proposing efficient techniques for processing the data streams. As the computing era of data models are changing very fast, what was considered as real time data processing in the previous era is not a real time today.

TABLE II
Data formats supported by current ETL techniques

Structured data	Unstructured data
Xiaofang Li [5]	Srividya K. [6]
F. Majeed [10]	J. Villanueva Chávez [7]
J. Song [11]	L. Jiang [8]
Shao YiChuan [12]	Marie-Aude Aufaure [9]
Kakish Kamal [13]	
Revathy S. [14]	
A. Wibowo [15]	
Marcin Gorawski [16]	
Alfredo Cuzzocrea [17]	
Alfredo Cuzzocrea [18]	
Imane Lebdaoui [19]	
M. Obalı [20]	
Rui Jia [21]	
L. Imane [22]	

TABLE II shows the data formats that the current ETL techniques support while processing the data. If we look at the data processing from another perspective, big data is not only about the quantum of data. As we consider multiple data sources which can produce data streams in different formats such as flat files, xml files, databases and so on, the big data are always characterized by heterogeneity. Therefore there is a need to address the issues that arise due to heterogeneity while processing the data streams. But if look at TABLE II, we can say that very few of the existing techniques are attempting to address the heterogeneity of the data streams.

V. CONCLUSION

Real time ETL techniques that are capable of handling data streams should consider the characteristics of the data streams such as volume, velocity and variety for efficient processing of the data. As the rate of data generated is increasing day by day, the data that was considered as real time few years back is no more a real time today with the expected response time decreasing. From this perspective, the existing ETL techniques partially address the issues related to the characteristics of big data. Therefore there is a need to fill the gap identified in this work and come up with a new solution to address the issues that arise due to these characteristics of the big data.

REFERENCES

[1] Gorawski, Marcin and Aleksander Chrószcz, "Query processing using negative and temporal tuples in stream query engines", CEE-SET, Heidelberg, Vol. 7054, pp. 70-83, Springer, 2012.

[2] Gorawski, Marcin, and Aleksander Chrószcz, "Synchronization modeling in stream processing", *Advances in Databases and Information Systems, Heidalberg, Vol. 186, pp. 91-102, Springer, 2013.*

[3] Gorawski M., "Advanced data warehouses", Habilitation, Studia Informatica 30(3B), 386, 2009.

[4] Babu, Shivnath, and Jennifer Widom, "Continuous queries over data streams", ACM SIGMOD Record, New York, Volume 30, Issue 3, pp. 109-120, ACM, September 2001.

[5] Xiaofang Li and Yingchi Mao, "Real-Time Data ETL Framework for Big Real-Time Data Analysis", ICIA, Lijiang, pp. 1289-1294, IEEE International Conference, 2015.

[6] Srividya K. Bansal and Sebastian Kagemann, Integrating Big Data: A Semantic Extract-Transform-Load Framework", in *Computer*, vol. 48, no. 3, pp. 42-50, IEEE, Mar. 2015.

[7] J. Villanueva Chávez and X. Li, "Ontology based ETL process for creation of ontological data warehouse", *CCE, 8th International Conference*, Merida City, pp. 1-6, IEEE, 2011.

[8] L. Jiang, H. Cai and B. Xu, "A Domain Ontology Approach in the ETL Process of Data Warehousing", *ICEBE, 7th International Conference*, Shanghai, pp. 30-35, IEEE, 2010.

[9] Marie-Aude Aufaure, Raja Chiky, Olivier Cure, Houda Khrouf, Gabriel Kepeklian, "From Business Intelligence to Semantic data stream management", *Future Generation Computer Systems, Vol. 63*, pp. *100-107,* Elsevier, *October 2016.*

[10] F. Majeed, Muhammad Sohaib Mahmood and M. Iqbal, "Efficient data streams processing in the real time data warehouse", *ICCSIT, 3rd IEEE International Conference*, Chengdu, pp. 57-60, IEEE, 2010.

[11] J. Song, Y. Bao and J. Shi, "A Triggering and Scheduling Approach for ETL in a Real-time Data Warehouse", *CIT, 10th International Conference*, Bradford, pp. 91-98, IEEE, 2010.

[12] Shao YiChuan, Xingjia Yao, "Research of Real-time Data warehouse Storage Strategy Based on Multi-level Caches", ICSSDMS, Macao, Vol. 25, pp. 2315-2321, ELSEVIER, April 2012.

[13] Kakish Kamal, and Theresa A. Kraft. "ETL evolution for real-time data warehousing", *Proceedings of the Conference on Information Systems Applied Research,* Vol. 2167, p. 1508, 2012.

[14] Revathy S., Saravana Balaji B. and N. K. Karthikeyan, "From Data Warehouse to Streaming Warehouse: A Survey on the Challenges for Real-Time Data Warehousing and Available Solutions", International Journal of Computer Applications, Vol. 81-no2, 2013.

[15] A. Wibowo, "Problems and available solutions on the stage of Extract, Transform, and Loading in near real-time data warehousing", *ISITIA,* Surabaya, pp. 345-350, IEEE, 2015.

[16] Marcin Gorawski and Anna Gorawska, "Research on the Stream ETL Process", BDAS, 10th International Conference, Poland, Vol. 424, pp. 61-71, Springer, 2014.

[17] Alfredo Cuzzocrea, Nickerson Ferreira and Pedro Furtado, "Enhancing Traditional Data Warehousing Architectures with Real-Time Capabilities", ISMIS, 21st International Symposium, Denmark, Vol. 8502, pp. 456- 465, Springer, 2014.

[18] Alfredo Cuzzocrea, Nickerson Ferreira and Pedro Furtado, "Real-Time Data Warehousing: A Rewrite/Merge Approach", LNCS, Germany, Vol. 8646, pp. 78-88, Springer, 2014.

[19] Imane Lebdaoui, Ghizlane Orhanou and Said Elhajji, "An Integration Adaptation for Real- Time Data Warehousing", IJSEIA, Vol. 8, pp. 115-128, 2014.

[20] M. Obalı, B. Dursun, Z. Erdem and A. K. Görür, "A real time data warehouse approach for data processing", *SIU,* Haspolat, pp. 1-4, IEEE, 2013.

[21] Rui Jia, Shicheng Xu and Chengbao Peng, "Research on Real Time Data Warehouse Architecture", ICICA, Singapore, Vol. 392, pp. 333-342, Springer, August 2013.

[22] LEBDAOUI Imane, Ghizlane ORHANOU, and Said EL HAJJI, "Data Integrity in Real-time Datawarehousing", *Proceedings of the World Congress on Engineering*, London, Vol. 3, 2013.