An Exploratory Study of Near-Real Time ETL Approaches for the Design of Agile Business Intelligence Infrastructure

Mohamed Minhaj
Associate Professor - Systems
SDMIMD, Mysuru
mminhaj@sdmimd.ac.in

Abstract

Most Business Intelligence (BI) systems used in organizations today leverage on the organization's data warehouses for harvesting business insights. The Extract, Transform and Load (ETL) process of Data Warehousing is critical in determining the querying, reporting or mining capabilities of BI systems. The demand for fresh data in data warehouses has always been a strong desideratum from the BI users but the conventional ETL process does not facilitate real time insights as the data in the warehouse is refreshed in an offline and batch mode. To address this issue, near-real time ETL approaches have emerged as a promising solution. This exploratory study, after examining the practical problems associated with the conventional ETL approach which are causing data latency in the Data Warehouses, endeavors to explore the key approaches devised towards near-real time ETL. This study also gives an account of key problems associated with the near- real time ETL approaches.

Keywords : *Business Intelligence, Data Warehousing, ETL*

Introduction

Business Intelligence (BI) Systems have been serving organizations in better decision making and improved performance by harvesting humongous amount of data that the organizations today have about their employees, suppliers, customers, their preferences etc. One of the key components of BI system is Data Warehouse. Data warehouses facilitate consolidation of data from a variety of sources and stored in heterogeneous formats in the organization. The querying or mining abilities of BI system depend on the efficiency of its data warehousing architecture, particularly the way in which the ETL (Extract, Transform and Load) process is performed.

The ETL process takes care of detecting relevant changes in the data from operational databases, extracts it into staging area, transforms it into appropriate formats and loads it into data warehouse. The ETL process, conventionally refreshes the data in the data warehouse in an offline and batch mode (Vassiliadis, 2009). This causes the level of freshness of data in the data warehouse not indicating the latest operational transactions and leads to an issue referred in the scholarly literature as Data Latency (Wibowo, 2015). To address this issue, near-real time ETL approaches have emerged as the most promising solution.

Objectives of the Study

The companies today capture trillions of bytes of information about their customers, suppliers, and operations. Millions of networked sensors are being embedded in the physical world in devices such as mobile phones, automobiles etc., for sensing, creating, and communicating data. This large pools of data which is generally referred as Big Data is now part of every sector and function of the global economy. Like other essential factors of production such as hard assets and human capital, it is increasingly the case that much of modern economic activity, innovation, and growth simply could not take place without data (Manyika, 2011). But the true value of this data comes from harvesting the right information at the right time. In the context of Business Intelligence, generally the ETL architecture determines the freshness, timeliness and efficiency of the data that is available in Data Warehouses for analysis. In view of the conventional ETL approaches used by data warehouses not facilitating real time data, this paper with help of the existing literature, endeavors to study the following:

1. The practical problems pertaining to the conventional ETL approach causing data latency in the Data Warehouses.
2. Elaborate on the motives that boost the need for near real time ETL in Business Intelligence Systems.
3. The different approaches devised for performing real time ETL and the key problems associated with real time ETL.

Business Intelligence (BI) Systems

The mention of the term “Business Intelligence” in scholarly literature dates back to 1865, wherein Richard

Millar Devensin used this term in his work to describe gaining profit by receiving and acting upon information about the environment. Also, in a 1958 article, IBM researcher Hans Peter Luhn used the term business intelligence to describe it as the ability to apprehend the interrelationships of presented facts in such a way as to guide action towards a desired goal (Luhn, 1958). Business intelligence as it is understood today is said to have evolved from the Decision Support Systems (DSS) that began in the 1960s and developed throughout the mid-1980s. DSS originated in the computer-aided models created to assist with decision making and planning. From DSS, data warehouses and business intelligence came into focus beginning in the late 80s. In 1989, Howard Dresner proposed "business intelligence" as an umbrella term to describe "concepts and methods to improve business decision making by using fact-based support systems." (Dssresources, 2014). Gartner defines BI as umbrella term that includes the applications, infrastructure and tools, and best practices that enable access to and analysis of information to improve and optimize decisions and performance. (Gartner, 2015)

The description of BI by Forrester Research is considered to be one of the widely used definition of BI. It defines BI as "A set of methodologies, processes, architectures, and technologies that transform raw data into meaningful and useful information used to enable more effective strategic, tactical, and operational insights and decision-making" .

The present business ecosystem is teeming with digital documents, images, audios, videos, and multiple forms of data that is unstructured or semi-structured in nature. Although conventionally, companies have been making business decisions based on transactional data stored in relational databases, in the recent years, they have realized that the non-traditional, less structured data in the form of weblogs, social media, email, sensors is trove of useful business insights. This phenomenon has fueled the growth of research and development in Business Intelligence. The current BI systems are capable of handling large amounts of unstructured data to help identify, develop and create new strategic business opportunities. The goal of BI is to allow easy interpretation of these large volumes of data. Identifying new opportunities and implementing an effective strategy based on insights can provide businesses with a competitive market advantage and long-term stability (Wikipedia, 2014).

The Role of Data Warehousing and ETL in BI Systems

Like all information systems, BI systems also have hardware, software, data, procedures and people. The key ingredients of BI systems are mining and reporting tools. These tools provide useful information to users who need it and when they need it. The information emanating from BI systems contain information representing patterns, relationships and trends about customers, suppliers, business partners and employees. Examples of BI systems include measuring and monitoring key performance indicators, benchmarking and forecasting sales, performing data mining and analysis of customer information to discover new business opportunities, and building enterprise dashboards to integrate and visualize information from various business areas. (Harvard Extension School, 2014)

BI applications generally use data gathered from a data warehouse and converts it into actionable insights. Data warehouse is a central repository of integrated data from many disparate sources. It is an enterprise wide storage of current and historical data used for analytical processing. The data stored in the warehouse is uploaded from the operational systems such as Marketing, HR etc. (Wikipedia, 2014). Typically enterprise data warehouse performs the following tasks:

Table 1 : Tasks performed by Data Warehouses

Congregates data from multiple sources into a single data store and enables a single query engine to analyze and present data.
Mitigates the problem of database isolation level lock contention in transaction processing systems caused by attempts to run large, long running, analysis queries in transaction processing databases.
Maintains data history, even if the source transaction systems does not.
Improve data quality by providing consistent codes and descriptions, flagging or even fixing bad data.
Presents the organization's information consistently.
Provides a single common data model for all data of interest regardless of the data's source.
Restructures the data so that it makes sense to the business users.

In the context of Data Warehousing, ETL refers to a core process that facilitates congregation of data from disparate sources and gives a single version of truth in an enterprise. ETL consists of three sub processes, namely – Extract, Transform and Load.

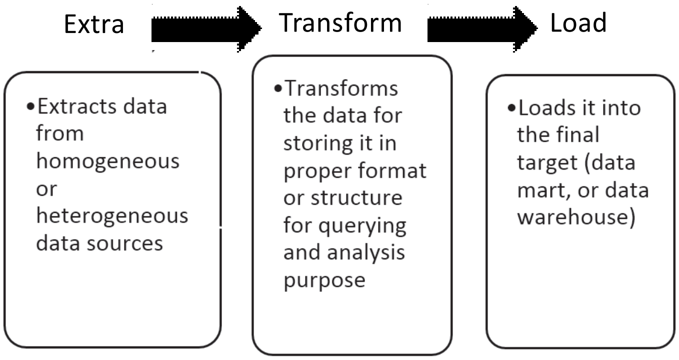
Extra
Transform
Load
•Extracts data from homogeneous or heterogeneous data sources
•Transforms the data for storing it in proper format or structure for querying and analysis purpose
•Loads it into the final target (data mart, or data warehouse)

Figure 1: Data warehouse's ETL process

All the three phases usually execute in parallel since the data extraction takes time, so while the data is being pulled, another transformation process executes, processing the already received data and prepares the data for loading and as soon as there is some data ready to be loaded into the target, the data loading starts without waiting for the completion of the previous phases.

One of the key challenges of ETL is that it has to extract and integrate data from multiple applications (systems), typically developed and supported by different vendors or hosted on separate computer hardware. Also, the disparate systems containing the original data are frequently managed by different employees and hence may vary in structures as well.

Transforming the data may involve the following tasks (Datawarehouse4u, 2015):

Table 2 : Typical tasks involved in Transform phase of ETL

Applying business rules (so-called derivations, e.g., calculating new measures and dimensions)
Cleaning (e.g., mapping NULL to 0 or "Male" to "M" and "Female" to "F" etc.)
Filtering (e.g., selecting only certain columns to load)
Splitting a column into multiple columns and vice versa
Joining together data from multiple sources (e.g., lookup, merge)
Transposing rows and columns
Applying any kind of simple or complex data validation (e.g., if the first 3 columns in a row are empty then reject the row from processing)

The load phase loads the data into the end target that is data warehouse or a data mart (A data warehouse that is limited in scope). Depending on the requirements of the organization, this process varies widely. Some data warehouses may overwrite existing information with cumulative information; updating extracted data is frequently done on a daily, weekly, or monthly basis. Other data warehouses (or even other parts of the same data warehouse) may add new data in a historical form at regular intervals.

Traditional Data Warehouse Architecture and the Associated Problems

A traditional data warehouse architecture consists of four layers: the data sources, the back-end, the global data warehouse, and the front-end (Vassiliadis, 2009). Typically, the data sources can be any of the following: On-Line Transaction Processing (OLTP) systems, legacy systems, flat files or files under any format. The set of operations taking place in the back stage of data warehouse architecture is generally known as the Extraction, Transformation, and Loading (ETL) processes. ETL processes are responsible for the extraction of data from different, distributed, and often, heterogeneous data sources, their cleansing and customization in order to fit business needs and rules, their transformation in order to fit the data warehouse schema, and finally, their loading into data warehouse. The global data warehouse keeps a historical record of data that result from the transformation, integration, and aggregation of detailed data found in the data sources. Moreover, this layer involves data stores that contain highly aggregated data, directly derived from the global warehouse (e.g., data marts and views.) The front-end level of the data warehouse architecture consists of applications and techniques that business users use to interact with data stored in the data warehouse.

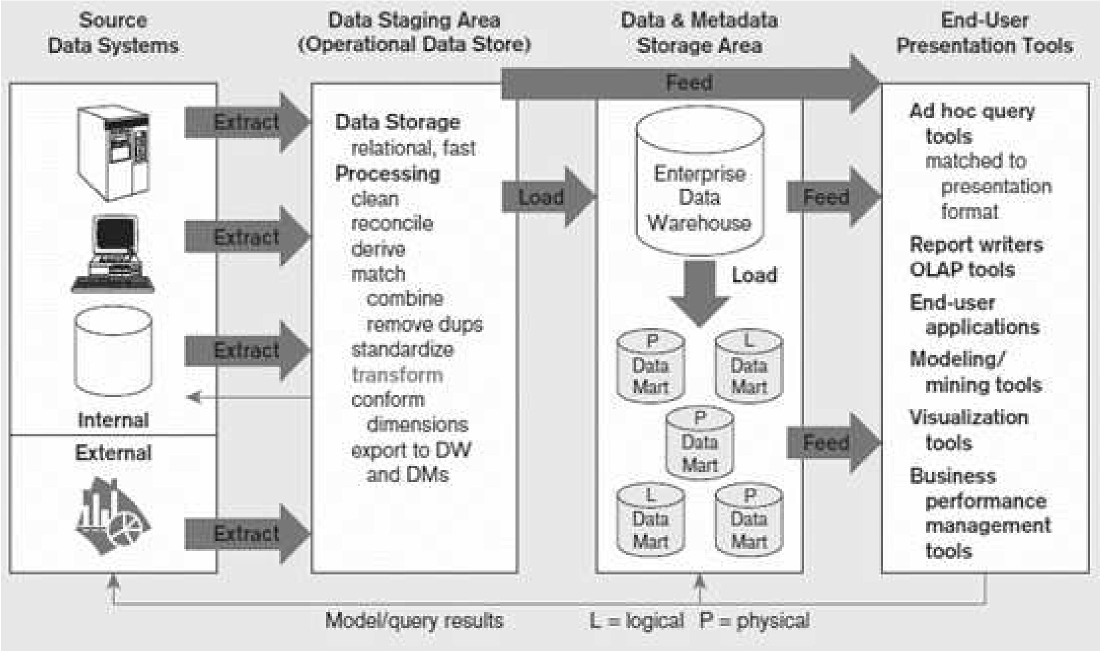

Figure 2: Traditional Data Warehouse Architecture

Traditionally, ETL processes deal with the following generic categories of problems:

Large volumes of data - The volumes of operational data are extremely large, and incur significant data management problems in all three phases of an ETL process.

Data quality - The data are not always clean and have to be cleansed.

Evolution of data stores - The evolution of the sources and the data warehouse can eventually lead even to daily maintenance operations.

Performance issues -. The whole process has to take place within a specific time window and it is necessary to optimize its execution time. In practice, the ETL process periodically refreshes the data warehouse during idle or low-load periods of its operation, e.g., every night. Any failures of the process must also be compensated within the specified time windows.

The Need for Near Real Time ETL

Data warehouse is updated through ETL (Extract, Transform and Loading) process. ETL has the responsibility to detect relevant data changes, extract it into staging area, transform it into appropriate format, and load it into data warehouse. Traditionally, ETL updates data warehouse periodically. This implies that data warehouse is not relevant to the current condition, where there is real time data between two updating process. Thus, it makes less accurate analysis result (Wibowo, 2015). The other problem is that traditional ETL should be performed at off peak hours. It means operational and analysis activities must be stopped, which would have repercussion on the systems that are expected to be running 24X7. Based on these problems, there should be a mechanism for updating the data warehouse immediately after the change in the source data, so that the enterprise's needs related to the latest data can be met. This requirement calls for a different approach of ETL to facilitate loading of data into data warehouse in a continuous manner unlike the periodic manner used in the traditional ETL approach.

The General Architecture for the near real time ETL process

According to the general architecture of a near real time ETL proposed by Panos Vassiliadis (Vassiliadis, 2009), the data warehouse consists of the following elements:

1) Data Sources hosting the data production systems that populate the data warehouse.

2) An intermediate Data Processing Area (DPA) where the cleaning and transformation of the data takes place and

3) The Data Warehouse (DW).

Each source can be assumed to comprize a data store (legacy or conventional) and an operational data management system (e.g., an application or a DBMS.) Changes that take place at the source side have first to be identified as relevant to the ETL process and subsequently propagated towards the warehouse, which typically resides in a different host computer. As per this architecture, each source hosts a Source Flow Regulator module that is responsible for the identification of relevant changes and propagates them towards the warehouse at periodic or convenient intervals, depending on the policy chosen by the administrators. This period is significantly higher than the one used in the current state-of-practice and has to be carefully calculated on the basis of the source system's characteristics and the user requests for freshness.

Also, the Data Processing Flow Regulator module is responsible of deciding which source is ready to transmit data. Once the records have left a certain source, an ETL workflow receives them at the intermediate data processing area. The primary role of the ETL workflow is to cleanse and transform the data in the format of the data warehouse. In principle, though, apart from these necessary cleansings and transformations, the role of the data processing area is versatile: (a) it relieves the source from having to perform these tasks, (b) it acts as the regulator for the data warehouse, too (in case the warehouse cannot handle the online traffic generated by the source) and (c) it can perform various tasks such as check pointing, summary preparation, and quality of service management. However, it is expected that a certain amount of incoming records may temporarily resort to appropriate Reservoir modules, so that the DPA can meet the throughput for all the workflows that are hosted there. Once all the ETL processing is over, data are ready to be loaded at the warehouse. A Warehouse Flow Regulator orchestrates the propagation of data from the DPA to the warehouse based on the current workload from the part of the end users posing queries and the QoS “contracts” for data freshness, ETL throughput and query response time.

Many other alternative architectures for near real time ETL have been suggested, some of the prominent approaches have been presented below:

Enterprise Application Integration (EAI) : These approaches have the ability to link transactions across multiple systems through existing applications by using software and computer systems architectural principles to integrate a set of enterprise computer applications. An EAI system is a push system, not appropriate for batch transformations, whose functionality entails a set of adapter and broker components that move business transactions in the form of messages across the various systems in the integration network. An adapter creates and executes the messages, while a broker routes messages, based on publications and subscription rules. The main benefit from an EAI system is fast extraction of relevant data that must be pushed towards the data warehouse. In general, an EAI solution offers great real time information access among systems, streamlines business processes, helps raise organizational efficiency, and maintains information integrity across multiple systems. Usually, it is considered as a good solution for applications demanding low latency reporting and bidirectional synchronization of dimensional data between the operational sources and the data warehouse. However, as nothing comes without a cost, they constitute extremely complex software tools, with prohibitively high development costs, especially for small and mid-sized businesses. Also, EAI implementations are time consuming, and need a lot of resources. Often, many EAI projects usually start off as point-to-point efforts, but very soon they become unmanageable as the number of applications increase.

Fast transformations via Capture - Transform - Flow (CTF) processes: This solution resembles a traditional ETL process too. CTF approaches simplify the real time transportation of data across different heterogeneous databases. CTF solutions move operational data from the sources, apply light-weight transformations, and then, stage the data in a staging area. After that, more complex transformations are applied (triggered by the insertions of data in the staging area) by microbatch ETL and the data are moved to a real time partition and from there, to static data stores in the data warehouse. CTF is a good choice for near real time reporting, with light integration needs and for those cases where core operations may share periods of low activity and due to that, they allow the realization of data synchronization with a minimal impact to the system.

Fast loading via micro batch ETL . This approach uses the idea of real time partitioning and resembles traditional ETL processes, as the whole process is executed in batches. The substantial difference is that the frequency of batches is increased, and sometimes it gets as frequent as hourly. Several methods can be used for the extraction of data - e.g., timestamps, ETL log tables, DBMS scrapers, network sniffers, and so on. After their extraction the data are propagated to the real time partition in small batches and this process continuously runs. When the system is idle or once a day, the real time partitions populate the static parts of the data warehouse. The microbatch ETL approach is a simple approach for real-time ETL and it is appropriate for moderate volumes of data and for data warehouse systems tolerant of hourly latency. The main message it conveys, though, is mainly that dealing with new data on a record-by-record basis is not too practical and the realistic solution resolves to finding the right granule for the batch of records that must be processed each time.

On-demand reporting via Enterprise Information Integration (EII) : EII is a technique for on-demand reporting. The user collects the data he needs on-demand via a virtual integration system that dispatches the appropriate queries to the underlying data provider systems and integrates the results. EII approaches use data abstraction methods to provide a single interface for viewing all the data within an organization, and a single set of structures and naming conventions to represent this data. In other words, EII applications represent a large set of heterogeneous data sources as a single homogenous data source. Specifically, they offer a virtual real time data warehouse as a logical view of the current status in the OLTP systems. This virtual warehouse is delivered on-the-fly through inline transformations and it is appropriate for analysis purposes. It generates a series of (SQL) queries at the time requested, and then it applies all specified transformations to the resulting data and presents the result to the end user. EII applications are useful for near-zero latency in real time reporting, but mostly for systems and databases containing little or no historical data.

Key Problems Associated with Near Real Time ETL

1. Problems related to extraction stage (Wibowo, 2015)

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 has infrequent updating process. Data stream is data that is not used repeatedly and is constantly changing.

b. Data source overload: Reading data source continuously can overload it and disturb the operational activities.

2. Problems associated with transformation stage.

a. Master data overhead: Data stored in data warehouse can be divided into two parts, namely master data and transactional data. Master data is the data that is not frequently changed. For example, product or customer. In the data warehouse, master data is implemented by dimension table. Transaction data changes frequently according to the transactions that occurs in data source. For example, 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. This problem is called by master data overhead.

b. Need of 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. In the near real time data warehousing, each data warehouse refreshment process only carries one or several small amounts of data. This results in the transformation process not being performed on each data warehouse refreshment cycle.

3. ***Problems associated with loading stage include Performance degradation and 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 called OLAP Internal inconsistency and this problem can occur during any OLAP operation such as roll up, drill down etc.

Concluding Remarks

The data in organizations is exponentially growing and the need for timely and accurate insights from the Business Intelligence (BI) systems used in many organizations is also constantly increasing. BI vendors are adopting agile practices to harvest mountains of organization's data to facilitate the right information to the right people at the right time. As most BI architectures harvest information stored in Enterprise Data Warehouses, the ability of BI to effectively harness and make sense of the disparate data depends on the efficiency of the ETL process used. The traditional approach of ETL is not relevant in the changing landscape of business. Today's businesses demand information that is as fresh as possible as the value of the business insights decreases as it gets older. Plethora of approaches have been devised to make the ETL process near-real time. But considering the contemporary dimensions of Business Intelligence Systems, the unique architectural requirements of Data Warehouses and optimization issues associated with ETL process, there is future scope for further research and development of better approaches towards real-time ETL.

References

A Brief History of Decision Support Systems. (2014). Retrieved 11 March 2016, from http://dssresources.com/history/dsshistory.html

Business Intelligence - BI - Gartner IT Glossary. (2015). Retrieved 11 March 2016, from http://

www.gartner.com/it-glossary/business-intelligence-bi/

Business intelligence - Wikipedia, the free encyclopedia. (2014). Retrieved 11 March 2016, from https://en.wikipedia.org/wiki/Business_intelligence#cite_note-1

Data & Analytics by KPMG (2015). Retrieved 11 March 2016, from http://www.kpmg.com/us/en/topics/data-analytics/pages/default.aspx

Data warehouse - Wikipedia, the free encyclopedia. (2015). Retrieved 11 March 2016, from https://en.wikipedia.org/wiki/Data_warehouse

ETL. (2015). Retrieved 11 March 2016, from http://datawarehouse4u.info/ETL-process.html

Gartner Magic Quadrant for Data Integration Tools. (2015). Retrieved 11 March 2016, from http://now.informatica.com/en_data-integration-magic-quadrant_tools_analyst-report_2942.html

Harvard Extension School. (2014). Retrieved 11 March 2016, from http://www.extension.harvard.edu/

How Much Data is Out There? - Webopedia.com. (2014). Retrieved 11 March 2016, from http://www.webopedia.com/quick_ref/just-how-much-data-is-out-there.html

Luhn, H. P. (1958). A business intelligence system. *IBM Journal of Research and Development, 2*(4), 314-319.

Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., & Byers, A. H. (2011). Big data: The next frontier for innovation, competition, and productivity.

Vassiliadis, P., & Simitsis, A. (2009). Near real time ETL. In *New trends in data warehousing and data analysis* (pp. 1-31).Springer US.

Wibowo, A. (2015, May). Problems and available solutions on the stage of Extract, Transform, and Loading in near real-time data warehousing (a literature study). In *Intelligent Technology and Its Applications(ISITIA), 2015 International Seminar on* (pp. 345-350). *IEEE.*