A Survey on Traditional and Cloud Based Data Warehousing Systems

Dr. Nilamadhab Mishra
Post Graduate Teaching & Research Dept., 09 School of Computing,
Debre Berhan University, Debre Berhan 445, Ethiopia.

Abstract-*A traditional data warehouse is a central store of data that has been extracted from operational data sources. Data in a data warehouse is typically subject-oriented, non-volatile, and of a historic nature, as contrasted with data used in an on-line transaction processing system. Data in data warehouses are often used in data mining and on-line analytical processing tools. Online analytical processing techniques do not process enterprise data for hidden or unknown intelligence. With the obsolescence of traditional data warehouse, new emerging technologies are progressively integrated in order to gain a better return on investment at enterprise level. In this paper, I discuss in details about the traditional data warehouse system and an emerging data warehouse system in cloud based vicinities in order to provide the better storage for large scale disparate data.*

Keywords-cloud based DW, traditional DW, data extraction, financial analysis, OLAP

I. INTRODUCTION

Data warehouses offer organizations the ability to gather and store enterprise data in a single conceptual enterprise repository. Basic data modelling techniques are applied to create relationship associations between individual data elements or data element groups. These associations, or "models," often take the form of entity relationship diagrams (ERDs). More advanced techniques are used that include star schema and snowflake schema model. Regardless of the technique chosen, the goal is to build a metadata model that conceptually represents the information usage and relationships within the enterprise [1], [2].

Leveraging the metadata model, enterprise users can then apply elementary data analysis techniques to gather business knowledge. For example, ad hoc queries can be run against the data warehouse to extract enterprise-level information. These queries would supply information that was impossible to obtain under the legacy system of disparate information [3], [4], [5].

More advanced data warehouse toolsets incorporate the concept of multidimensional data, or data cubes. This data structure allows information to be multi-indexed, which allows for rapid drill-down on data attributes. Data cubes are usually used to perform what-if scenarios over identified data indices. For example, suppose Company X sells jewellery and has offices in Detroit, Pittsburgh, and Atlanta. If the proper attributes were chosen as indices, a user could perform the following analysis.

i) What was the enterprise's total revenue for 2001?
ii) What was Atlanta's revenue in November?
iii) If there were a 30 percent increase in orders during the first quarter of 2002, what would my year-end revenue be for Pittsburgh?
iv) If the Detroit office were closed, what would the impact be to the bottom line?

This multidimensional analysis of multiple business views is called Online Analytical Processing (OLAP). The primary function of OLAP systems is to provide users the ability to perform manual exploration and analysis of enterprise summary and detailed information. It is important to understand that OLAP requires the user to know what information he or she is searching for. OLAP techniques do not process enterprise data for hidden or unknown intelligence [6], [7], [8], [9], [10].

During the mid- to late 1990s, commercial vendors began exploring the feasibility of applying traditional statistical and artificial intelligence analysis techniques to large databases for the purpose of discovering hidden data attributes, trends, and patterns. This exploration evolved into formal data-mining toolsets based on a wide collection of statistical analysis techniques.

For a commercial business, the discovery of previously unknown statistical patterns or trends can provide valuable insight into the function and environment of their organization. Data-mining techniques allow businesses to make predictions of future events, whereas OLAP only gives an analysis of past facts. Data-mining techniques can generally be grouped into one of three categories: clustering, classifying, and predictive.

Clustering techniques group information based on a set of input patterns using an unsupervised or undirected

algorithm. One example of clustering could be the analysis of business consumers for unknown attribute groupings. Input to this example would be well-defined consumer attributes over which the algorithm would search.

Classifying techniques group or assign objects to predetermined groupings based on well-defined attributes. The groupings are often clusters discovered using the above techniques. An example would be assigning a consumer to a particular sales cluster based on their income level.

Predictive techniques take as input known attributes regarding a particular object or category and apply those attributes to another similar group to identify expected behaviour or outcomes. For example, if a group of individuals wearing helmets and shoulder pads is known to be a football team, we can expect another group of individuals with helmets and pads to be a football team as well.

The following list describes many data-mining techniques in use today. Each of these techniques exists in several variations and can be applied to one or more of the categories above.

Regression modelling—this technique applies standard statistics to data to prove or disprove a hypothesis. One example of this is linear regression, in which variables are measured against a standard or target variable path over time. A second example is logistic regression, where the probability of an event is predicted based on known values in correlation with the occurrence of prior similar events.

Visualization—this technique builds multidimensional graphs to allow a data analyst to decipher trends, patterns, or relationships.

Correlation—this technique identifies relationships between two or more variables in a data group.

Variance analysis—this is a statistical technique to identify differences in mean values between a target or known variable and nondependent variables or variable groups.

Discriminate analysis—this is a classification technique used to identify or "discriminate" the factors leading to membership within a grouping.

Forecasting—forecasting techniques predict variable outcomes based on the known outcomes of past events.

Cluster analysis—this technique reduces data instances to cluster groupings and then analyses the attributes displayed by each group.

Decision trees—Decision trees separate data based on sets of rules that can be described in "if-then-else" language.

Neural networks—neural networks are data models that are meant to simulate cognitive functions. These techniques learn through iteration over data, allowing for greater flexibility in the discovery of patterns and trends.

The rest of this paper is organized as follows. Section II discusses the data extraction approach in traditional data warehousing system. Section III discusses the cloud based data warehousing approach. Section IV discusses a comparison analysis between Traditional and cloud based approaches. Finally section V concludes this paper.

II. DATA EXTRACTION APPROACHES IN TRADITIONAL DATA WAREHOUSES (TDW)

Extraction is the operation of extracting data from a source system for further use in a data warehouse environment. This is the first step of the ETL process. After the extraction, this data can be transformed and loaded into the data warehouse. The source systems for a data warehouse are typically transaction processing applications. For example, one of the source systems for a sales analysis data warehouse might be an order entry system that records all of the current order activities [11], [12], [13], [14].

Designing and creating the extraction model is a time-consuming task in the entire data warehousing process. The source systems might be very complex and poorly documented, and thus determining which data needs to be extracted can be difficult. The data has to be extracted normally not only once, but several times in a periodic manner to supply all changed data to the data warehouse and keep it up-to-date. Moreover, the source system typically cannot be modified, nor can its performance or availability be adjusted, to accommodate the needs of the data warehouse

These are important considerations for extraction and ETL in general. This chapter, however, focuses on the technical considerations of having different kinds of sources and extraction methods. It assumes that the data warehouse team has already identified the data that will be extracted, and discusses common techniques used for extracting data from source databases.

Designing this process means making decisions about the following two main aspects:

- Which extraction method do I choose?

This influences the source system, the transportation process, and the time needed for refreshing the warehouse.

- How do I provide the extracted data for further processing?

This influences the transportation method, and the need for cleaning and transforming the data.

The extraction method you should choose is highly dependent on the source system and also from the business needs in the target data warehouse environment. Very often, there is no possibility to add additional logic to the source systems to enhance an incremental extraction of data due to the performance or the increased workload of these systems. Sometimes even the customer is not allowed to add anything to an out-of-the-box application system.

The estimated amount of the data to be extracted and the stage in the ETL process (initial load or maintenance of data) may also impact the decision of how to extract, from a logical and a physical perspective. Basically, you have to decide how to extract data logically and physically.

A. Logical Extraction Approaches

There are two approaches of logical extraction: Full Extraction and Incremental Extraction.

In case of full extraction, the data is extracted completely from the source system. Because this extraction reflects all the data currently available on the source system, there's no need to keep track of changes to the data source since the last successful extraction. The source data will be provided as-is and no additional logical information (for example, timestamps) is necessary on the source site. An example for a full extraction may be an export file of a distinct table or a remote SQL statement scanning the complete source table.

In case of incremental extraction, at a specific point in time, only the data that has changed since a well-defined event back in history will be extracted. This event may be the last time of extraction or a more complex business event like the last booking day of a fiscal period. To identify this delta change there must be a possibility to identify all the changed information since this specific time event. This information can be either provided by the source data itself such as an application column, reflecting the last-changed timestamp or a change table where an appropriate additional mechanism keeps track of the changes besides the originating transactions. In most cases, using the latter method means adding extraction logic to the source system [15], [16], [17], [18], [19], [20].

Many data warehouses do not use any change-capture techniques as part of the extraction process. Instead, entire tables from the source systems are extracted to the data warehouse or staging area, and these tables are compared with a previous extract from the source system to identify the changed data. This approach may not have significant impact on the source systems, but it clearly can place a considerable burden on the data warehouse processes, particularly if the data volumes are large.

B. Physical Extraction Methods

Depending on the chosen logical extraction method and the capabilities and restrictions on the source side, the extracted data can be physically extracted by two mechanisms. The data can either be extracted online from the source system or from an offline structure. Such an offline structure might already exist or it might be generated by an extraction routine.

There are the following methods of physical extraction: Online Extraction and Offline Extraction.

In case of online extraction, the data is extracted directly from the source system itself. The extraction process can connect directly to the source system to access the source tables themselves or to an intermediate system that stores the data in a preconfigured manner (for example, snapshot logs or change tables). Note that the intermediate system is not necessarily physically different from the source system. With online extractions, you need to consider whether the distributed transactions are using original source objects or prepared source objects.

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

Following structures should be considered:

Flat files- Data are defined in a generic format. Additional information about the source object is necessary for further processing.
Dump files-Oracle-specific format. Information about the containing objects may or may not be included, depending on the chosen utility.

Redo and archive logs-Information is in a special, additional dump file.

In this section, I discussed the data extraction approaches in traditional data warehousing system.

III.CLOUD BASED DATA WAREHOUSE APPROACH (CBDW)

A key factor driving the evolution of the modern data warehouse is the cloud. So, it is called as a cloud based data warehouse (CBDW) or data warehouse as a cloud based service. This CBDW creates access to near infinite low-cost storage, improved scalability, outsourcing of DW management, and provides security to the cloud vendor. It has potential to pay for only the storage and computing resources actually used. CBDW brings data revolution & provides more people to access the data driven insights. CBDW enables organizations to focus on what they want to achieve using software rather than on the management of that software and associated hardware [21], [22]. The main motivation of CBDW is instant availability of resources with minimal management and without any physical infrastructure at enterprise level. Through instant availability, the DW application is available for use within hours or even seconds of when a customer purchases it, rather than in weeks or months. Hardware and DW software are also deployed, configured, and managed by the application provider as part of the service. The minimal management tells that users do not spend time worrying about how to patch, upgrade, scale, and optimize the DW software, because the software service does that automatically.

The service is ready to go at any time. There's no setup overhead and no need to spend time doing capacity planning, procuring and installing hardware, installing and configuring software, or any other prep work. Data is entered into the system with minimal worry. There's no time spent transforming data to get it into a form that the DW can handle. The focus is on what questions to ask about data. The service does not require manual tuning and configuration changes to deliver performance and efficiency.

It all just works, without interruptions. There's no need to worry about ongoing management of the system, data protection, security or other concerns because these are handled by the service. It adapts dynamically and immediately to changes. The service monitors and observes the data warehouse and adapts, identifying and making optimizations based on how the service is being used [23], [24].

In CBDW, the business users interact with the integrated development environment through the business applications. The data security, backup and recovery, application hosting, and infrastructure scalability can be effectively maintained through the same integrated development environment by the cloud developers [25], [26], [27].

IV.COMPARISON ANALYSIS CONCERNING TDW AND CBDW

David floyer in February 2016 has analysed the financial growth between TDW and CBDW across the business enterprises [28], [29], [30]. Net Present Value (NPV) is the difference between the present value of cash inflows and the present value of cash outflows. The NPV in CBDW is three times more than the NPV in CBDW. IRR-is the interest rate at which the net present value of all the cash flows. The IRR in CBDW is approximately seven times more than the IRR in TDW. Also, the break-even point (BEP) analysis states that if the enterprise implements a TDW, then after twenty six months the enterprise starts gaining benefits by recovering the expanses that has been made during TDW setup and tuning, where as in CBDW, only four months are required to meet the said expanses. The analysis states that the overall financial growth of CBDW at enterprise level is much higher than TDW (figure-1).

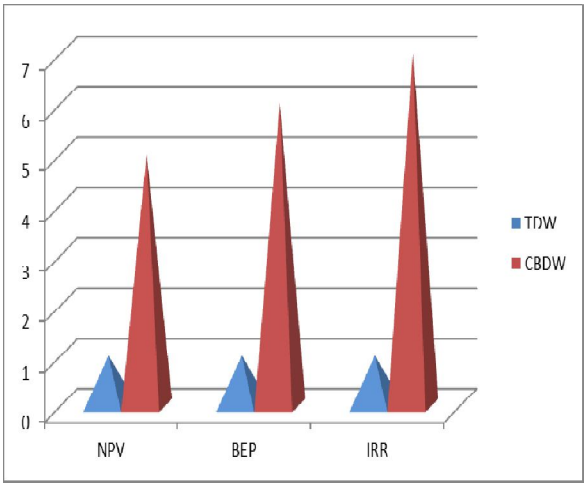

Figure-1: financial growth rate analysis in between TDW and CBDW.

V. CONCLUSIONS

In this paper, a discussion is prepared on the overall aspects of traditional and cloud based data warehousing systems. There is a rapid invocation of cloud based data warehouse across many business enterprises due to high financial growth as compared to traditional data warehouse. In cloud based data warehouse, many new technologies are progressively integrated to handle the disparate data sources that consist of sensor data, IoT data, RFID data, business data, social sensing data, smart dust data, and the data generated from other computing devices.

ACKNOWLEDGMENT

The author would like to express thanks to the Post Graduate Teaching & Research Dept., at School of Computing, Debre Berhan University, Ethiopia for supporting this research.

REFERENCES

[1] Mishra, N., Lin, C. C., & Chang, H. T. (2014). Cognitive inference device for activity supervision in the elderly. The Scientific World Journal, 2014.

[2] Kimball, R. (1998). The data warehouse lifecycle toolkit: expert methods for designing, developing, and deploying data warehouses. John Wiley & Sons.

[3] Mishra, N., Lin, C. C., & Chang, H. T. (2015). A cognitive adopted framework for IoT big-data management and knowledge discovery prospective. International Journal of Distributed Sensor Networks, 11(10), 718390.

[4] Cui, Y., & Widom, J. (2003). Lineage tracing for general data warehouse transformations. The VLDB Journal—The International Journal on Very Large Data Bases, 12(1), 41-58.

[5] Mishra, N., Lin, C. C., & Chang, H. T. (2014, December). A cognitive oriented framework for IoT big-data management prospective. In Communication Problem-Solving (ICCP), 2014 IEEE International Conference on (pp. 124-127). IEEE.

[6] Chang, H. T., Mishra, N., & Lin, C. C. (2015). IoT Big-Data Centred Knowledge Granule Analytic and Cluster System for BI Applications: A Case Base Analysis. PloS one, 10(11), e0141980.

[7] Mishra, N., Chang, H. T., & Lin, C. C. (2014). Data-centric knowledge discovery strategy for a safety-critical sensor application. International Journal of Antennas and Propagation, 2014.

[8] Inmon, W. H., Zachman, J. A., & Geiger, J. G. (1997). Data stores, data warehousing and the Zachman framework: managing enterprise knowledge. McGraw-Hill, Inc.

[9] Mishra, N., Chang, H. T., & Lin, C. C. (2015). An Iot knowledge reengineering framework for semantic knowledge analytics for BI-services. Mathematical Problems in Engineering, 2015.

[10] Ross, T. R., Ng, D., Brown, J. S., Pardee, R., Hornbrook, M. C., Hart, G., & Steiner, J. F. (2014). The HMO Research Network Virtual Data Warehouse: a public data model to support collaboration. Egems, 2(1).

[11] Mishra, N. (2011). A Framework for associated pattern mining over Microarray database. International Journal of Global Research in Computer Science (UGC Approved Journal), 2(2).

[12] Padmanabhan, R., & Patki, A. U. (2016). U.S. Patent No. 9,430,505. Washington, DC: U.S. Patent and Trademark Office.

[13] Mishra, N., Chang, H. T., & Lin, C. C. (2018). Sensor data distribution and knowledge inference framework for a cognitive-based distributed storage sink environment. International Journal of Sensor Networks, 26(1), 26-42.

[14] Lavdas, Steve, and Doug McDowell. "Methods and apparatus for improving data warehouse performance." U.S. Patent 8,738,576, issued May 27, 2014.

[15] Mishra N, (2017). "In-network Distributed Analytics on Data-centric IoT Network for BI-service Applications", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN: 2456-3307, Volume 2, Issue 5, pp.547-552, September-October.2017.

[16] De Bus, L., Diet, G. U. N. T. E. R., Gadeyne, B., Leroux-Roels, I., Claeys, G., Steurbaut, K., ... & Depuydt, P. (2014). Validity analysis of a unique infection surveillance system in the intensive care unit by analysis of a data warehouse built through a workflow-integrated software application. Journal of Hospital Infection, 87(3), 159-164.

[17] Patnaik, B. C., & Mishra, N. (2016). A Review on Enhancing the Journaling File System. Imperial Journal of Interdisciplinary Research, 2(11).

[18] Chang, H. T., Li, Y. W., & Mishra, N. (2016). mCAF: a multi-dimensional clustering algorithm for friends of social network services. SpringerPlus, 5(1), 757.

[19] Zhong, R. Y., Huang, G. Q., Lan, S., Dai, Q. Y., Chen, X., & Zhang, T. (2015). A big data approach for logistics trajectory discovery from RFID-enabled production data. International Journal of Production Economics, 165, 260-272.

[20] Chang, H. T., Liu, S. W., & Mishra, N. (2015). A tracking and summarization system for online Chinese news topics. Aslib Journal of Information Management, 67(6), 687-699.

[21] Khnaisser, C., Lavoie, L., Diab, H., & Ethier, J. F. (2015, September). Data warehouse design methods review: trends, challenges and future directions for the healthcare domain. In East European Conference on Advances in Databases and Information Systems (pp. 76-87). Springer, Cham.

[22] Peral, J., Ferrández, A., De Gregorio, E., Trujillo, J., Maté, A., & Ferrández, L. J. (2015). Enrichment of the phenotypic and genotypic Data Warehouse analysis using Question Answering systems to facilitate the decision making process in cereal breeding programs. Ecological informatics, 26, 203-216.

[23] Kapoor, Rahul, Gauray Rewari, Renu Chintalapati, Aravind Sridharan, Ravishanka Muniasamy, Florian Schouten, David Shenk, and Srinivas M. Vedagiri. "Automated Definition of Data Warehouse Star Schemas." U.S. Patent Application 14/921,972, filed April 27, 2017.

[24] Lopes, C. C., Times, V. C., Matwin, S., Ciferri, R. R., & de Aguiar Ciferri, C. D. (2014, September). Processing OLAP queries over an encrypted data warehouse stored in the cloud. In International Conference on Data Warehousing and Knowledge Discovery (pp. 195-207). Springer, Cham.

[25] Hashem, I. A. T., Yaqoob, I., Anuar, N. B., Mokhtar, S., Gani, A., & Khan, S. U. (2015). The rise of "big data" on cloud computing: Review and open research issues. Information Systems, 47, 98-115.

[26] Cuzzocrea, A., Moussa, R., Xu, G., & Grasso, G. M. (2015, May). Cloud-based OLAP over big data: Application scenarios and performance analysis. In Cluster, Cloud and Grid Computing (CCGrid), 2015 15th IEEE/ACM International Symposium on (pp. 921-927). IEEE.

[27] Szczerba, M., Wiewiórka, M. S., Okoniewski, M. J., & Rybiński, H. (2016). Scalable Cloud-Based Data Analysis Software Systems for Big Data from Next Generation Sequencing. In Big Data Analysis: New Algorithms for a New Society (pp. 263-283). Springer, Cham.

[28] Floyer, D. (2016). The Vital Role of Edge Computing in the Internet of Things.(2016).

[29] Kelly, J., Floyer, D., Vellante, D., & Miniman, S. (2014). Big data vendor revenue and market forecast 2012-2017. Wikibon. org.

[30] Floyer, D. (2014). Enterprise Big-data [Online] Available: http://wikibon. org/wiki/v.