Research in Big Data Warehousing using Hadoop

ABSTRACT

Traditional data warehouses have played a key role in decision support system until the recent past. However, the rapid growing of the data generation by the current applications requires new data warehousing systems: volume and format of collected datasets, data source variety, integration of unstructured data and powerful analytical processing. In the age of the Big Data, it is important to follow this pace and adapt the existing warehouse systems to overcome the new issues and challenges. In this paper, we focus on the data warehousing over big data. We discuss the limitations of the traditional ones. We present its alternative technologies and related future work for data warehousing.

Keywords: warehousing, big data, Hadoop, Hive

INTRODUCTION

Nowadays, one of the most important and difficult challenges in software systems research is to develop software and tools for storage, manage, and handling information on large amounts of data. Currently, a lot of this data is stored in a non-structured manner, using different languages and formats. Because of their large size and complexity of this data, managing the data with traditional approaches is not suitable. Hadoop is an alternative solution for the next generation data volumes. It is characterized by a high capacity, fault-tolerance, scalability, parallel processing and easily manageable. Although Hadoop (Hadoop, 2016) is the best known for Map-Reduce and its distributed file system (HDFS), the term is also used for a family of related projects that fall under the umbrella of infrastructure for distributed computing and large-scale data processing (White, 2012). As the Hadoop ecosystem grows, more projects are appearing, not necessarily hosted at Apache. Such projects provide complementary services to Hadoop and add higher-level abstractions. Among these projects, Hive is a data warehousing solution developed by the Facebook team. Apache Hive (hive, 2016) allows the managing and interrogating data without requiring the writing of Map-Reduce programs which are hard to maintain and reuse (Thusoo et al., 2010).

The contributions of this paper can be summarized as follows: (1) We give the traditional data warehouse limitations, which have led to the appearance of the modern data warehouse. (2) We present Hadoop and Hive Platforms and some related work in the context of Data Warehousing over these platforms. (3) A critical discussion about the limitations of new platforms and their open research issues in the context of Data Warehousing will be given.

The remainder of this paper is organized as follows: Section 2 provides background information about traditional data warehousing, its limitations. Section 3 describes the Hadoop and Hive Platforms and some related work of data warehousing over Hadoop. Section 4 provides the Hadoop-based data warehousing limitations and the open research issues of future work. The last section concludes this work.

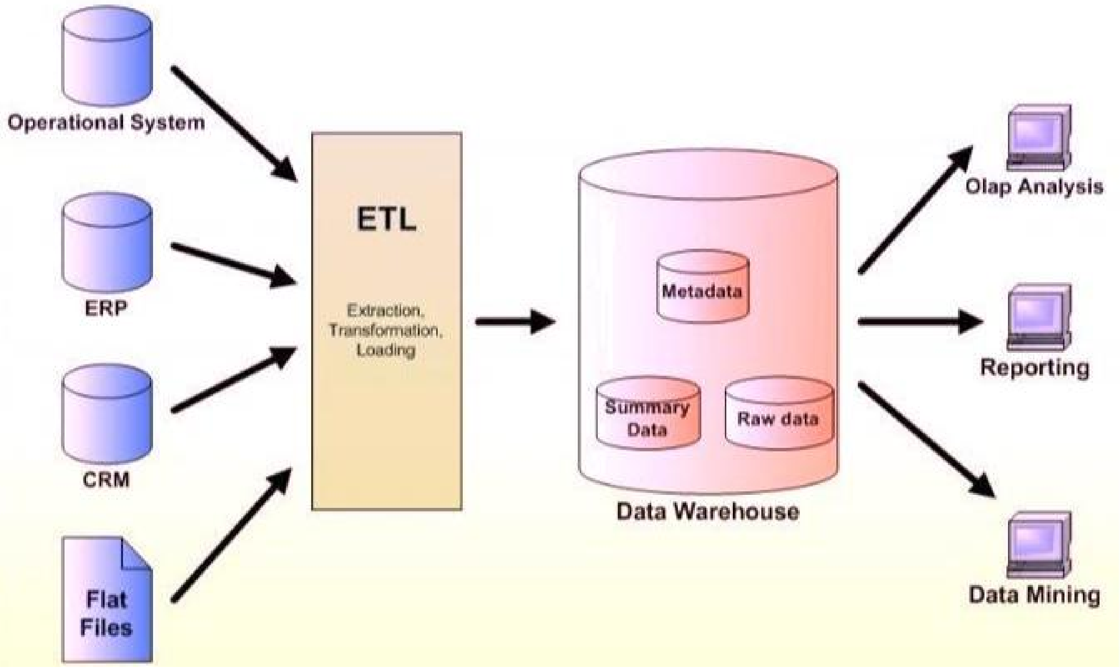

Figure 1. General architecture of a data warehouse (Inmon, 2005)

TRADITIONAL DATA WAREHOUSES

Traditional DW Concepts

"A data warehouse (DW) is a collection of data, organized to be used as a decision support", where data are organized by themes or subjects", (Inmon, 2005). For Example, Production, Sales, Marketing, etc. This organization allows gathering all the information related to a specific topic in order to facilitate decision-making. Data in a warehouse are mainly used in consultation mode, and are less frequently modified or deleted by users; this keeps the traceability of information in order to perform analysis over a long period. Data integration in a DW eliminates all conflicts of representation, names, and context, to get a consistent and a uniform representation of data when data are loaded at the DW (Shah et al., 2005). **Figure 1** illustrates the general architecture of a DW. It stores data from various sources of heterogeneous and distributed information. These sources may include databases, data files, external sources, etc. Before being stored, data sources must first be cleaned. The cleaning process is to select and purify the data to eliminate problems and reconcile the semantic differences between these data. Once cleaned, the data will be integrated into the warehouse.

'The ETL (extract, transform, and load) process is used to integrate data from multiple sources, it is necessary to perform the loading of data sources at the logical warehouse schema transformations' (Golfarelli, 2010). This is done in three steps: (1) Extraction which is to retrieve data from several sources. This step requires synchronization of the extraction process to ensure data integrity. (2) Transformation is to format the extracted data according to the target warehouse schema using a set of rules. For example, assigning semantics to the data sources and combining the source fields to target fields. (3) Loading is to load data into a target database, DW or a data mart to be analyzed. Information about creation, management, and use of the warehouse is stored in a separate directory in the warehouse. This information called "*Metadata*" contains information about the schema sources, the corresponding data, the integration schema, refresh rules, user profiles and user groups (Coronel et al., 2013). A data Warehouse may be composed of several data Marts. They are excerpts from the warehouse dedicated to one type of users and meet a specific need. They are dedicated to the OLAP (Online Analytical Processing) analysis and decision making. OLAP provides multidimensional views of data to support tools decision (White, 2012).

Traditional DW Limitations

Several data warehouses have been developed in different fields. For instance (Sebaa et al., 2015; Sebaa et al., 2016; Shah et al., 2005). However, today's DWs are faced with new scientific challenges. Indeed, the current data sources are heterogeneous, autonomous, scalable, and distributed. With these challenges, the traditional data warehouses are faced some limitations, which are summarized by Krishnan (2013) with the following sentence "Lack of scalability due to processing complexities coupled with inherent data issues and limitations of the underlying hardware, application software, and other infrastructure", and that we detailed as follows:

Data nature: New models and formats of semi-structured and unstructured data have created a need to be integrated and used by the modern data warehouses however traditional DW can not handle semi-structured unstructured data.

Data availability : The unavailability of data at the appropriate time affects the use and adoption of decision making within the company.

Storage mechanism and architecture: using the same set of disks and controllers in the data warehouse creates a huge penalty on both the availability and system performance.

Query Performance: The analytic and ad-hoc queries cause the greatest impact on overall query performance, access, and processing of data, including moving through the network because of their non-deterministic. Queries can require a large set of data that needs to access multiple storage areas or a large data set that is available in a smaller storage area.

New business requirements, and organic or inorganic growth of the business that requires a new data management methods and new warehouse architecture.

HADOOP BASED DATA WAREHOUSE

Hadoop

Apache Hadoop (Hadoop, 2016) is the open-source implementation of Google's Map-Reduce computing model. Hadoop is based on The HDFS (Hadoop Distributed File system) (hdfs, 2016), a scalable, robust and fault-tolerant distributed file system that provides high-throughput access to application data. Hadoop provides a distributed storage, and parallel processing across clusters using the Map-Reduce paradigm. It automatically replicates and collocates data across multiple nodes. Thus, Hadoop lowers the cost of infrastructure. It has been adopted early by several big companies, (Facebook, Yahoo, Amazon, Adobe, etc.).

Hive

As described by Thusoo and al. (Thusoo et al., 2010), Apache Hive (hive, 2016) is an open-source data warehousing solution built on top of Hadoop. Under Hive, the queries are expressed in a declarative language SQL-like, called HiveQL, which are compiled into map-reduce jobs that are executed using Hadoop. Hive supports primitive and complex type. Hive stores data in tables, where each table consists of a number of rows and each row consists of a specified number of columns. It processes data for analysis, not to serve users, so it does not need ACID guarantees (as for traditional relational database) for data storage or retrieval (Bronson et al., 2015). According to Thusoo and al. (Thusoo et al., 2010), Hive has improved the performance of Hadoop itself by ~20%.

Why Hadoop and Hive as Modern Data Warehouse

The existing software and hardware can not follow anymore the constant increasing of the data volume which exceeds their ability to process them. However, through its components (Hive and others), the Hadoop ecosystem enables developers to focus on solving their Big data warehousing problems. Indeed, several parameters push us to choose Hadoop:

- Hadoop is a low-cost storage repository.
- It is a platform that supports running ETL processes in parallel.
- It allows pushing the processing to the data rather than data to the processing.
- With Hadoop, raw data is loaded directly to low-cost commodity servers one time, and only the higher value refined results are passed to other systems.

Big Data Warehouse Projects

The Hadoop and Hive projects have a diverse and growing community of both users researchers and industrials, as will be seen from the large number of projects described below:

Social network field. As detailed in (Bronson et al., 2015), Hive is Facebook's data warehouse, with 300 petabytes of data in 800,000 tables. Facebook generates 4 new Petabytes of data and runs 600,000 queries and 1 million map-reduce jobs per day. At Facebook, data grow by millions of events (inserts) per second and process tens of petabytes and hundreds of thousands of queries per day. However, some limitations and challenges arise on Facebook DW such as, (1) How trading off storage space against CPU time? Since Hive uses standard compression techniques to save space, but this requires more time processing. (2) How minimize data replication across data centers?

Medical field. To Improving the prediction of traumatic brain injury survival rates, Rodger (2015) used Hadoop and Hive to orchestrate database processing by marshaling the distributed servers, running the various tasks in parallel. They collected data on three ship variables (Byrd, Boxer, Kearsage) and injuries to four body regions (head, torso, extremities, and abrasions) to determine how the set of collected variables relates to the body injuries. The data analysis objective is the prevention of a traumatic brain injury. Hadoop-GIS (Aji et al., 2013) is a spatial data warehouse based on Hadoop for executing fast queries and handling complex spatial data on large volumes of spatial data. It optimizes, translates, and submits queries by the query language stored procedure (QLSP) . Under Hadoop-GIS, Data is partitioned by splitting the spatial data into subsets that can be processed in parallel across worker nodes.

Climatology field. Sinha (2016) presents an Apache Hadoop framework for analyzing climate datasets, where the data are generated from satellite images. Datasets structure of climate, such as NetCDF and HDF were designed for archiving data.

Energy field. EDF is the largest producer and supplier of electricity in France and in the world. This company has developed the Smart-grid project on which a lot of smart meters and sensors are deployed. The Apache Hadoop is used for storing massive time -series of the data deluge provoked by the smart meters. HIVE has been used for queries analysis and HBASE at the forefront of data access. This project has given new perspectives for energy management such as electric vehicles, and renewable energy generation (Picard, 2013).

Telecom field. SFR is the second largest telecommunication operator in France. It has used the Cloudera distribution of Hadoop framework in conjunction with its data warehouses. It has established a big data environment to reduce operating costs of its data. It has improved the knowledge of its customers' behavior and has implemented actions to retain them by analyzing data logs (Cloudera, 2016).

Physics field. The HEPDOOP framework (Bhimji et al., 2014) is implemented to analyze data workflow. It uses different tools as Pig and Hadoop for mass data processing and Python Scikit-Learn for multivariate analysis.

DISCUSSION AND FUTURE WORK

Hadoop and Hive are till now Frameworks in progress. They are open source projects and are actively enhanced by both industrial and searcher communities. In this section, we give the current limitations of the Apache Hive and Hadoop solutions to the data warehousing develop, the issues and promising directions that are likely to drive future work.

Hadoop and Hive Limitation

Hadoop can not handle updates (deletion, insertion, or updating a record) since it uses HDFS. HDFS is based on the Write Once-Read Many data access model (a file once created, written, and closed can not be updated again except for appending data to its end), which means as well that it does not support OLAP operations.

On the other hand, HiveQL the query language of Apache Hive also has its own limitations. Indeed, it does not support many standard SQL queries, such as update queries of data and type, delete queries, sub-queries, real-time queries, and transactions. Therefore, it can not be used as OLTP tool. Hive is not adapted for small data sets since queries have higher latency, due to the start-up overhead for Map-Reduce jobs.

Future Work

To describe the future work we follow the typical data warehousing architecture. It includes data sources, storage system, OLAP Engine and front-end tools.

Data sources. New data sources should be considered such as Internet of Things. Internet of Things which is the development of the Internet towards a network of interconnected objects, ranging from houses, cars and transportation cargos to electrical appliances. The implementation of data warehouse over big data and IoT lead to a set of questions that need to be answered in future research, which include:

What are the appropriate models to be used for modelling the distributed stream data of IoT applications in the big data warehouse?

How to overcome the inherent complexity and data volume in order to provide the appropriate information to the user?

What are the appropriate extraction tools of IoT data sources?

Storage system. As we have seen in the previous subsection, whatever the performances and power that these new platforms have reached, there is still always lacking to be bridged.

Apache Hive must cover a wide range of query types which can not be handled until now. Thus, researchers and industrial communities need to work harder in this aspect.

Hadoop needs new mechanisms which will enable it to manage separate updates (deletion, insertion, or updating a record) in Hadoop Distributed File System.

OLAP Engine. As detailed by Cuzzocrea and al. (2013), data Warehousing and OLAP over Big Data need new methods of aggregations computing, computational paradigms, and designing OLAP data cubes considering the context of big data.

Relational and multi-dimensional models are the main data models used in data warehousing literature. However, these two models can not meet all needs of new data for analysis and business intelligence. Therefore, new data models which can support the complexity of big data warehousing are required.

Storing the huge amount of data requires defining new security and privacy policies over big data warehouse.

Front-end tools. A novel class of front-end tools and open solutions must be developed, in order to cope with emerging challenges posed by OLAP cubes over Big data.

CONCLUSION

In this paper, we have reviewed the data warehousing concepts, its architecture, and its limitations. Then, we have presented the new platforms (Hadoop and Hive) of modern data warehouses and some important related work. A critical discussion of the limitations of new platforms and their open research issues in the context of Data Warehousing is given.