ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 10, October 2015

A Study on Evolution of Data in Traditional RDBMS to Big Data Analytics

Surajit Mohanty[1], Kedar NathRout[2], Shekharesh Barik[3], Sameer Kumar Das[4]

Asst. Prof., Computer Science & Engineering, DRIEMS, Cuttack, India [1, 2, 3]

Asst. Prof., Computer Science & Engineering, GATE, Berhampur, India [4]

Abstract: The volume of data that enterprise acquires every day is increasing rapidly. The enterprises do not know what to do with the data and how to extract information from this data. Analytics is the process of collecting, organizing and analysing large set of data that is important for the business. The process of analysing and processing this huge amount of data is called bigdata analytics. The volume, variety and velocity of big data cause performance problems when processed using traditional data processing techniques. It is now possible to store and process these vast amounts of data on low cost platforms such as Hadoop. The major aspire of this paper is to make a study on data analytics, big data and its applications.

Keywords: BigData, Hadoop, MapReduce, Sqoop and Hive.

I. INTRODUCTION

The volume of data that enterprise acquires every day is increasing rapidly. In this way Traditional RDBMS fails to store huge amount of data. Up to GB of Data can be Stored in different verities of RDBMSs. It is not recommended to use RDBMS if volume of data increases to hexa byte of things. Even though It deals with GB of data, still it provides degradation of performance. Seek time is improving more slowly than transfer rate. Seeking is the process of moving the disk's head to a particular place on the disk to read or write data. It characterizes the latency of a disk operation, whereas the transfer rate corresponds to a disk's bandwidth. If the data access pattern is dominated by seeks, it will take longer to read or write large portions of the dataset than streaming through it, which operates at the transfer rate. On the other hand, for updating a small proportion of records in a database, a traditional B-Tree (the data structure used in relational databases, which is limited by the rate it can perform, seeks) works well. For updating the majority of a database, a B-Tree is less efficient than MapReduce, which uses Sort/Merge to rebuild the database. MapReduce can be seen as a complement to an RDBMS.

II. PRODUCTION OF BIG DATA

Big data is being generated by everything around us at all times. Every digital process and social media exchange produces it. Systems, sensors and mobile devices transmit it. Big data is arriving from multiple sources at an alarming velocity, volume and variety. To extract meaningful value from big data, you need optimal processing power, analytics capabilities. Now A days to handle big data, our traditional RDBMS fails, not able to store this large volume of data. [1] So Hadoop is the solution. So Bigdata is the problem and Hadoop is the solution. In other words it can be told as Bigdata is the issue, Hadoop is the implementation. For example Google is producing every day data up to more than 12 PB like Facebook providing 10 PB, ebay producing 8 PB per day.

For storing and processing of large volume of data we need use Hadoop as Framework. Hadoop is a framework used for storing and processing of large volume of data. Whereas traditional RDBMS can only store data, not able to process the data. For this we need to write more complex logic by following any programming Language. It's too tedious to write code for the same.

III. EVOLUTION OF MAPREDUCE TO PROGRAMMING LANGUAGE

MapReduce is a good fit for problems that need to analyse the whole dataset, in a batch fashion, particularly for ad hoc analysis. An RDBMS is good for point queries or updates, where the dataset has been indexed to deliver low-latency retrieval and update times of a relatively small amount of data. MapReduce suits applications where the data is written once, and read many times, whereas a relational database is good for datasets that are continually updated. [2]

Another difference between MapReduce and an RDBMS is the amount of structure in the datasets that they operate on. Structured data is data that is organized into entities that have a defined format, such as XML documents or database tables that conform to a particular predefined schema. This is the realm of the RDBMS. Semi-structured data, on the other hand, is looser, and though there may be a schema, it is often ignored, so it may be used only as a guide to the structure of the data: for example, a spreadsheet, in which the structure is the grid of cells, although the cells themselves may hold any form of data. Unstructured data does not have any particular internal structure: for example, plain text or image data.

MapReduce works well on unstructured or semi structured data, since it is designed to interpret the data at processing time. In other words, the input keys and values for MapReduce are not an intrinsic property of the data, but they are chosen by the person analysing the data. [5]

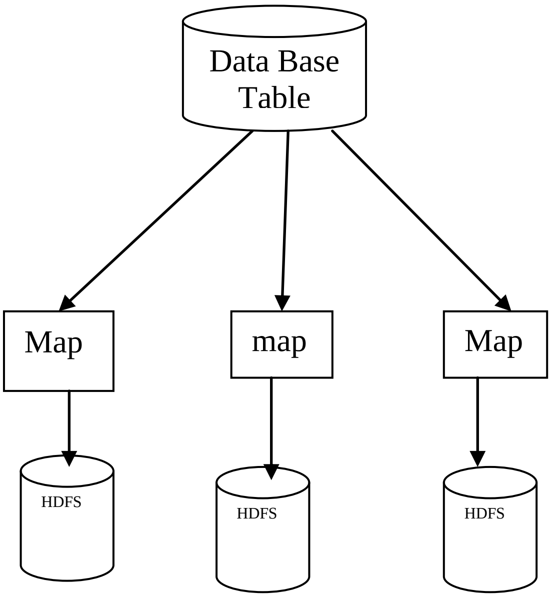
DOI 10.17148/IJARCCE.2015.41049

IV.REUSING TRADITIONAL RDBMS BY SQOOP

Sqoop is one of the component of Hadoop built on top of HDFS and is meant for interacting target RDMS only such as to import the data from RDBMS table to HDFS (Hadoop distributed File System) or to export the data from HDFS to any RDBMS table. [3]

Sqoop is only meant for importing and exporting the data to and from RDBMS, but it will never ever processing of hadoop data using business logic.

Sqoop is a command-line interface application for transferring data between relational databases and Hadoop. It supports incremental loads of a single table or a free form SQL query as well as saved jobs which can be run multiple times to import updates made to a database since the last import. Imports can also be used to populate tables in Hive or HBase. Exports can be used to put data from Hadoop into a relational database [6]



V. EVOLUTION OF HIVE OVER MAPREDUCE

Hive is one of the components of Hadoop built on top of HDFS and It is a warehouse kind of system in Hadoop stack. Hive is meant for querying of the data, advanced queries and for data summarisation. All the data Hive is going to be organised by means table only.
Whenever the sql kind queries that we are providing as part of Hive are been converted internally to corresponding Map Reduce jobs. Hive is introduced in Facebook and afterword it was opted by Apache Software Foundation.

VI. APPLICATION OF HADOOP

The volume of data that enterprise acquires every day is increasing rapidly. To store and mine this huge volume of data, Hadoop is a good framework. Hadoop provides a framework to process data of this size using a computing cluster made from normal, commodity hardware. There are two major components to Hadoop: the file system, which is a distributed file system that splits up large files onto multiple computers, and the MapReduce framework, which is an application framework used to process large data stored on the file system. Hadoop Distributed File System (HDFS) is the core technology for the efficient scale out storage layer, and is designed to run across low-cost commodity hardware. Apache Hadoop YARN is the pre-requisite for Enterprise Hadoop as it provides the resource management and pluggable architecture for enabling a wide variety of data access methods to operate on data stored in Hadoop with predictable performance and service levels.
YARN is a next-generation framework for Hadoop data processing extending MapReduce capabilities by supporting non-MapReduce workloads associated with other programming models.

VII. MAIN DISADVANTAGES OF HADOOP

a. Security Concerns
b. Vulnerable By Nature
c. Not Fit for Small Data
d. Potential Stability Issues

Using advanced analytics such as Hadoop to mine big data implementations in the enterprise has raised concerns about how to secure and control the data repositories. By distributing data storage and data management across a large number of nodes in an enterprises. Data volumes are doubling annually, and roughly 80 percent of that captured data is unstructured, and must be formatted using a technology like Hadoop in order to be mineable for information. Considering this growth, it is clear that security concerns won't be going away anytime soon.

VIII. CONCLUSION

To handle Bigdata, our traditional RDBMS fails, not able to store this large volume of data. So Hadoop is the solution. So Bigdata is the problem and Hadoop is the solution. In Other words it can be told as Bigdata is the issue, Hadoop is the implementation. Hadoop has moved far beyond its beginnings in web indexing and is now used in many industries for a huge variety of tasks that all share the common theme of lots of variety, volume and velocity of data i.e. both for structured and unstructured data. It is now widely used across industries, including finance, media and entertainment, government, healthcare, information services, retail, and other industries with Big Data requirements but the limitations of the original storage infrastructure remain.

REFERENCES

[1] OnurSavas, YalinSagduyu, Julia Deng, and Jason Li,Tactical Big Data Analytics: Challenges, Use Cases and Solutions, Big Data Analytics Workshop in conjunction with ACM Sigmetrics 2013,June 21, 2013
[2] DunrenChe, MejdlSafran, and Zhiyong Peng, From Big Data to Big Data Mining: Challenges, Issues, and Opportunities, DASFAA Workshops 2013, LNCS 7827, pp. 1–15, 2013
[3] Laurila, Juha K., et al. The mobile data challenge: Big data for mobile computing research. Proceedings of the Workshop on the Nokia Mobile Data Challenge, in Conjunction with the 10th International Conference on Pervasive Computing. 2012. https://research.nokia.com/files/public/MDC2012_Overview_LaurilaGaticaPerezEtAl.pdf

DOI 10.17148/IJARCCE.2015.41049

[4] Kaushik, Rini T., and Klara Nahrstedt. "T: a data-centric cooling energy costs reduction approach for big data analytics cloud." proceedings of the International Conference on High Performance Computing, networking, Storage and Analysis. IEEE Computer Society Press, 2012 http: //conferences. computer.org/ sc/2012 / papers/ 1000a037.pdf

[5] Jefry Dean and Sanjay Ghemwat,. MapReduce: Simplified data processing on large clusters, Communications of the ACM, Volume 51 pp. 107–113, 2008

[6] Marcin Jedyk, MAKING BIG DATA, SMALL, Using distributed systems for processing, analysing and managing large huge data sets, Software Professional's Network, Cheshire Data systems Ltd.

BIOGRAPHIES

Mr Surajit Mohanty, Assistant professor in Dept. of CSE, DRIEMS, Tangi, Cuttack. He has 8 year industry and teaching experience. He completed his M-tech in the year 2010.His area of interest is in SAP, ERP, Data mining and BIG data analysis.

Mr. Kedar NathRout, Assistant professor in Dept. of CSE, DRIEMS, Tangi, Cuttack since 20th Aug. 2008, completed his M-tech in the year 2010. His area of interest is in Java/J2EE, BIG data analysis.

Mr. Shekharesh Barik, Assistant professor in Dept. of CSE, DRIEMS, Tangi, Cuttack since 5th Aug. 2007, completed his M-tech in the year 2010. His area of interest is in computer graphics and BIG data analysis.

Mr. Sameer Kumar Das working as an Asst. professor in cse at GATE, Berhampur completed his M.Tech in computer science and engineering 2011. His area of interest is in computer networking and BIG data analysis.

Copyright to IJARCCE
DOI 10.17148/IJARCCE.2015.41049