Big Data and Big Data Scrutiny with Hadoop's MapReduce

Ms. Nikita P. Rane[1]
Student
Department of Computer Science and Engineering
Shri Sant Gadge Baba College of Engineering and Technology, Bhusawal.

Prof. Dinesh D. Patil[2]
Associate Professor and Head
Department of Computer Science and Engineering
Shri Sant Gadge Baba College of Engineering and Technology, Bhusawal.

Abstract

Big Data is highly related to large-volume of data, complex with evolving relationship, growing data sets with multiple, heterogeneous and self-governing sources. There is a faster development of networking along with data storage and collection capacity. The data is said as "Big Data" due to its characteristics of Volume, Variety, Velocity and Veracity. Most of this Big Data is unstructured, semi structured and heterogeneous in nature. The volume and the heterogeneity of Big Data, with the speed it is generated, make it difficult for the present computing infrastructure to manage Big Data. Because of this nature of Big Data, traditional data management, warehousing and analysis systems are not satisfactorily able to analyze this data. In order to process Big Data, HACE Theorem is considered that characterizes the features of Big Data for Big Data Processing. Hadoop and HDFS by Apache is a software framework which is widely used for storing, managing and analyzing Big Data which is a challenging task as it involves large distributed file systems which should be fault tolerant, flexible and scalable. Hadoop's MapReduce is widely being used for the efficient processing of large data sets on clusters which is nothing but Big Data. In this paper, the various solutions are introduced in hand through Map Reduce framework over Hadoop Distributed File System (HDFS). Map Reduce is a Minimization technique which makes use of file indexing with mapping, sorting, shuffling and finally reducing. Map Reduce techniques have been introduced which is implemented for Big Data analysis using HDFS.

Index Terms: Big Data, Big Data Analysis, HACE Theorem, HDFS, Map Reduce.

1. Introduction

Now-a-days, data from the web for example: data from Social Sites is generated in Exabyte's (10^{18}), Zettabyte's (10^{21}) which is nothing but Big Data. Social Media, Public Picture sharing site, data from science and research are generating vast amount say of size 2.5 quintillion bytes of data daily. This above said data is nothing but Big Data. Big Data is a collection of large datasets that are so large and complex that it becomes difficult to capture, process and store it using traditional data processing applications. Big Data generally includes large data sets with sizes that cannot be easily operated on by traditional software tools to capture, curate, process this Big Data with the tolerable elapsed time that is within the required time. Data size is constantly changing that leads to Big Data and this is one of the challenging tasks. Big Data arrives from different heterogeneous, autonomous sources with complex and have continuously evolving relationship.

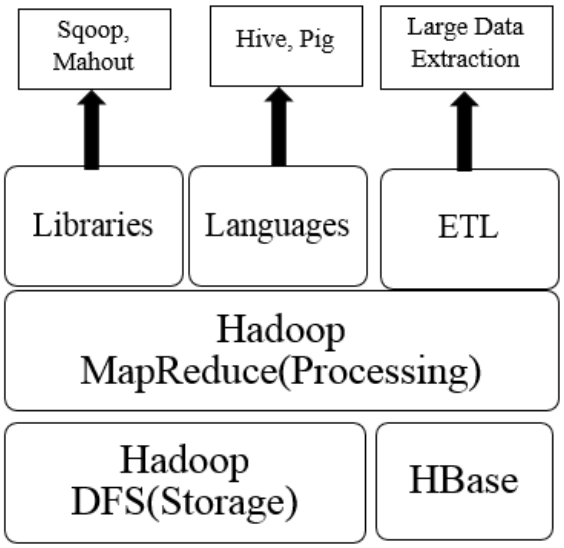
HACE Theorem [1] has been proposed that characterizes different features in order to process Big Data. These characteristics plays a challenging role in order to retrieve knowledge from Big Data. Big Data is a combination of structured (data that is contained in relational databases and spreadsheets that is in a predefined format and predefined length), semi-structured (a form of structured data, but doesn't have a formal structure, also entities in the same group can have different attributes, for example: email- abc.p@ffg.mn) and unstructured data (data that are not organized in a predefined format for example: image, audio, video, metadata). Instead of RDBMS, NoSQL that refers to Not Only SQL like Apache Hadoop can be used to meet the needs of Big Data. NoSQL databases are unstructured in nature and hence can be used to process structured as well as unstructured data. As said before Big Data is a combination of structured that is well-formatted, semi-structured, a form of structured data and unstructured that is in user query form data, analysis of Big Data becomes a challenging issue. In order to process Big Data, it requires proper data storage and analysis, proper Big Data processing which requires parallel processing with distributed systems, query processing and fault tolerant systems.

- *Ms. Nikita P. Rane is currently pursuing masters degree program in computer science and engineering in North Maharashtra University, India, E-mail: nprane.it@gmail.com*
- *Prof. Dinesh D. Patil, Associate Professor and Head in compter science and engineering in North Maharashtra University, E-mail: dineshonly@gmail.com*

2. The Crux of Apache Hadoop

Apache Hadoop is an open source framework that supports distributed processing of large data sets, also it allows to work with thousands of nodes and petabytes of data. Hadoop is designed to scale up the processing from a single cluster to multiple number of clusters such that each cluster providing processing and storage. Hadoop is a scalable, fault tolerant, reliable shared storage.

Apache Hadoop consist of two core components, namely: HDFS (Hadoop Distributed File System) and MapReduce. Basically, HDFS deals with the storage of Big Data and MapReduce deals with the processing of the Big Data. Basic Hadoop architecture as shown in Hadoop architecture shows the different layers of Hadoop system, where the bottom layer if HDFS layer,



Hadoop Architecture

which provides a scalable and reliable data storage on a large number of clusters of commodity servers. HDFS divides the block of data into uniform sizes of 64 MB or 128 MB. In HDFS layer, which is culpable for data storage, data is arranged into files and directories. HDFS layer is responsible for corruption detection and recovery. Following are the features of HDFS:

- Provides reliable storage
- Provides fault tolerant systems as prepares multiple redundant copies of data sets.
- Provides fast access
- Provides write-once-multiple-reads access model

HDFS has master slave architecture. HDFS components are as follows:

- NameNode

- DataNode
- Secondary DataNodes

NameNode serves as a master node, whereas DataNode serves as a slave node.

- NameNode manages the filesystem namespace, such as directories and files, along with this also manages the data blocks which are present on DataNodes. Also manages block redundancy.
- DataNodes are the slaves that are set-up with multiple machines. These slaves are actually responsible for storing the data blocks and to handle read-write requests to the stored data.
- Secondary NameNode is culpable for carrying out checkpoints periodically. In case of, NameNode failure, the NameNode that is the master, can be restarted using these checkpoints.

The layer above the HDFS layer in Hadoop architecture is MapReduce layer. MapReduce supports parallel and distributed processing. MapReduce is a parallel programming model which is responsible for parallel processing of large data sets on a cluster. MapReduce layer is also presented in a master slave architecture. MapReduce is responsible for Batch processing. The data from the storage of HDFS layer is divided uniformly and is distributed on multiple nodes. MapReduce is responsible for communication between multiple nodes, splitting and scheduling of task, fault tolerance, monitoring and reporting.

The various components of MapReduce are as follows:

- JobTracker
- TaskTracker
- JobHistoryServer

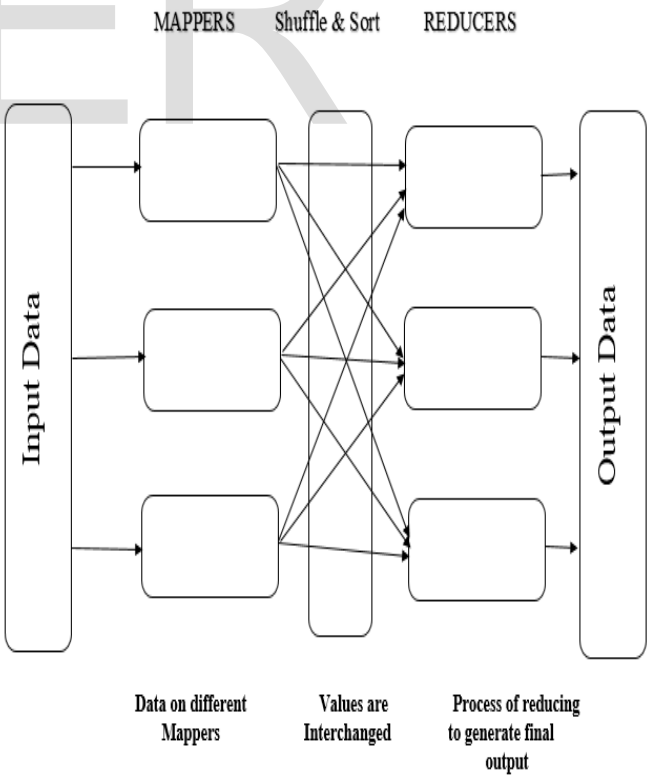
1) JobTracker serves as a master of the system which is responsible to manage the jobs, resources and communication between nodes in the cluster. This master component is responsible for scheduling the jobs to the slaves, monitoring them and re-executing the task which has failed.
2) TaskTracker is responsible for running the task assigned by the master i.e. by JobTracker. The task can be of any type, it can be a map task or a reduce task.
3) JobHistoryServer is responsible for servicing all job history, which is related to the queries from the users.

MapReduce operates in three phases:

- Map Phase
- Shuffle and Sort Phase
- Reduce Phase

1) Map Phase: Map phase is the initial phase where Map() function is applied to the partition data from HDFS and the output is written to the temporary storage. It is responsible for retrieving something which is related to the query from the record. Each record in a format of <key, value> pair.
2) Shuffle and Sort Phase: The main task of this phase is to assign all values with same intermediate key to same Reducer.
3) Reduce Phase: The reduce phase is responsible to summarize, aggregate, filter or transform all the results from different mapper's.

The basic information flow in shown below in the MapReduce programming model is that the relevant input data, i.e. the input split from HDFS, is read. The <key, value> pair is analyzed and the output is passed to the Map() function. This function processes the pairs and produces the intermediate pairs. The partition function is defined and executed, which stores the key-value pairs in the local storage. When the partition function completes the storage, it informs the master that the task is complete and the location where the data is stored. The master then passes all the information to the workers of the Reduce phase, which becomes an input to the Reduce phase. When all the intermediate values are retrieved, they are sorted according to their key. All the records with the same key are grouped together. Then, a Reduce() function is executed with input <key, group_of_values>. The Reduce phase then carries out processing on input by performing Reduce() function and produces the final output. The output is then associated with the local file system and is made available to HDFS. Following diagram shows the basic structure of MapReduce Programming Mode.



Hadoop's MapReduce Processing Model

3. Big Data Scrutiny with Hadoop MapReduce

Now-a-days, Big Data processing has become an important issue due to its multisource, massive, heterogeneous and dynamic characteristics. HACE Theorem [1] was introduced to consider the different characteristics while processing the Big Data. These characteristics begins with large-volume of data, alongwith **H**eterogeneous, **A**utonomous sources with decentralized and distributed control, to extract **C**omplex and **E**volving relationships between the data. The Big Data processing, analysis requires parallel programming modes like MapReduce. A general-purpose parallel programming model [2] has been proposed which consist of parallel processing on large set of machine-learning algorithms for multicore-processors, which is based on MapReduce parallel programming model. The basic idea behind this technique to expedite the speed of machine learning applications just by applying multiple cores at the problem rather than optimizing the problem. Ten different classical algorithms are accomplished in the framework, namely, Locally weighted linear regression, Neural Network, Naïve Bayes, Logistic Regression, Gaussian Discriminative Analysis, K-means, Principle Component Analysis, Expectation Maximization, Independent Component Analysis, Support Vector Machine. The main motive [2] is to find how above mentioned data mining machine learning algorithms can be transformed into a summation form in a MapReduce framework to improve the processing, in order to improve the speed of machine learning algorithms by using multiple cores. Summation form allows to speed up the different cores. Summation operation can be applied to different large scale data that has been separated into several subsets. A fast Parallel K-Means clustering algorithm [3] which is based upon MapReduce parallel programming mode is used in order to scale-up, speed-up the large datasets processing effectively.

4. Conclusion

Currently, Big Data processing has become an important issue as the data is growing fast in petabytes and zettabytes. Big Data processing mainly depends on parallel programming models. The HACE theorem gives various characteristics for Big Data analysis which is essential so as to consider different aspects while processing Big Data. Big Data processing requires parallel processing models like Apache Hadoop's MapReduce. This Apache Hadoop's

MapReduce allows parallel processing of large data sets in a less computational time. MapReduce as it allows parallel and distributed processing plays a vital role in data mining with Big Data. Big Data processing can be performed effectively using Apache Hadoop's MapRdeuce but the performance can be more effective when MapReduce is used in combination to HDFS. Big Data processing which comprises of unstructured and structured data can be processed effectively and efficiently using Hadoop.

References

[1] Xindong Wu,Gong-Qing Wu, Wei Ding,Data Mining With Big Data,IEEE transactions on Knowledge and data engineering, vol 26, no 1,January 2014.

[2] C.T. Chu, S.K. Kim, Y.A. Lin, Y. Yu, G.R. Bradski, A.Y. Ng, and K. Olukotun, "Map-Reduce for Machine Learning on Multicore," Proc. 20th Ann. Conf. Neural Information Processing Systems (NIPS '06), pp. 281-288, 2006.

[3] Weizhong Zhao, Huifang Ma, Qing He, "Parallel K-Means Clustering Based on MapReduce", *Springer-Verlag Berlin Heidelberg*, pp. 674-679, 2009.

[4] "IBM" What Is Big Data: Bring Big Data to the Enterprise, "http://www-01.ibm.com/software/data/bigdata/ , IBM, 2012".