ISSN 2349-6193
Impact Factor: 2.805

IJESMR

International Journal OF Engineering Sciences & Management Research

SURVEY PAPER ON BIG DATA ANALYTICS AND HADOOP

Ms. Rucha S Sheloadkar[1] and Prof. Himanshu U Joshi[2]

[1]Department of Computer Engineering, JSPM's Imperial College of Engineering and Research, Pune, India

[2]Department of Computer Engineering, JSPM's Imperial College of Engineering and Research, Pune, India

Keywords: Big Data, Hadoop, MapReduce, HDFS.

ABSTRACT

Data are now woven into every sector and function in the global economy, and, like other essential factors of production such as hard assets and human capital, much of modern economic activity simply could not take place without them. The use of Big Data — large pools of data that can be brought together and analyzed to discern patterns and make better decisions — will become the basis of competition and growth for individual firms, enhancing productivity and creating significant value for the world economy by reducing waste and increasing the quality of products and services. Big Data demand cost-effective, fault tolerant, scalable and flexible and innovative forms of information processing for decision making. This paper emphasis on the features, architectures, and functionalities of Big data, Hadoop,Map Reduce, HDFS.

INTRODUCTION

As the information technology spreads fast, most of the data were born digital as well as exchanged on internet today. According to the estimation of Lyman and Varian [1], the new data stored in digital media devices have already been more than 92 % in 2002, while the size of these new data was also more than five exabytes. In fact, the problems of analyzing the large scale data were not suddenly occurred but have been there for several years because the creation of data is usually much easier than finding useful things from the data.Even though computer systems today are much faster than those in the 1930s,the large scale data is a strain to analyze by the computers we have today.

In response to the problems of analyzing large-scale data, quite a few efficient methods [2], such as sampling, data condensation, density-based approaches, grid-based approaches, divide and conquer, incremental learning, and distributed computing, have been presented. Of course, these methods are constantly used to improve the performance of the operators of data analytics process.1 the results of these methods illustrate that with the efficient methods at hand, we may be able to analyze the large-scale data in a reasonable time. The dimensional reduction method is a typical example that is aimed at reducing the input data volume to accelerate the process of data analytics. Another reduction method that reduces the data computations of data clustering is sampling [4], which can also be used to speed up the computation time of data analytics.

Although the advances of computer systems and internet technologies have witnessed the development of computing hardware following the Moore's law for several decades, the problems of handling the large-scale data still exist when we are entering the age of big data. That is why Fisher et al. [5] pointed out that big data means that the data is unable to be handled and processed by most current information systems or methods because data in the big data era will not only become too big to be loaded into a single machine, it also implies that most traditional data mining methods or data analytics developed for a centralized data analysis process may not be able to be applied directly to big data. In addition to the issues of data size, Laney [6] presented a well-known definition (also called 3Vs) to explain what the "big" data is: volume, velocity, and variety. The definition of 3Vs implies that the data size is large, the data will be created rapidly, and the data will be existed in multiple types and captured from different sources, respectively. Later studies [7, 8] pointed out that the definition of 3Vs is insufficient to explain the big data we face now. Thus, veracity, validity, value, variability, venue, vocabulary, and vagueness were added to make some complement explanation of big data [8].

BIG DATA

It is important to realize that big data comes in many shapes and sizes. It also has many different uses – real-time fraud detection, web display advertising and competitive analysis, call center optimization, social media and sentiment analysis, intelligent traffic management and smart power grids, to name just a few. All of these analytical solutions involve significant (and growing) volumes of both multi-structured and structured data. Many of these analytical solutions were not possible previously because they were too costly to implement, or

ISSN 2349-6193
Impact Factor: 2.805

IJESMR

International Journal OF Engineering Sciences & Management Research

because analytical processing technologies were not capable of handling the large volumes of data involved in a timely manner. In some cases, the required data simply did not exist in an electronic form.

New and evolving analytical processing technologies now make possible what was not possible before. Examples include:

• New data management systems that handle a wide variety of data from sensor data to web and social media data.

• Improved analytical capabilities (sometimes called advanced or big analytics) including event, predictive and text analytics.

• Faster hardware ranging from faster multi-core processors and large memor spaces, to solid-state drives and tiered data storage for handling hot and cold data.

Supporting big data involves combining these technologies to enable new solutions that can bring significant benefits to the business.

In big data challenges induced by traditional data generation, consumption, and analytics at a much larger scale, newly emerged characteristics of big data has shown important trends on mobility of data, faster data access and consumption, as well as ecosystem capabilities. Fig. 1 illustrates a general big data network model with Map Reduce. Distinct applications in the cloud has put demanding requirements for acquisition, transportation and analytics of structured and unstructured data.

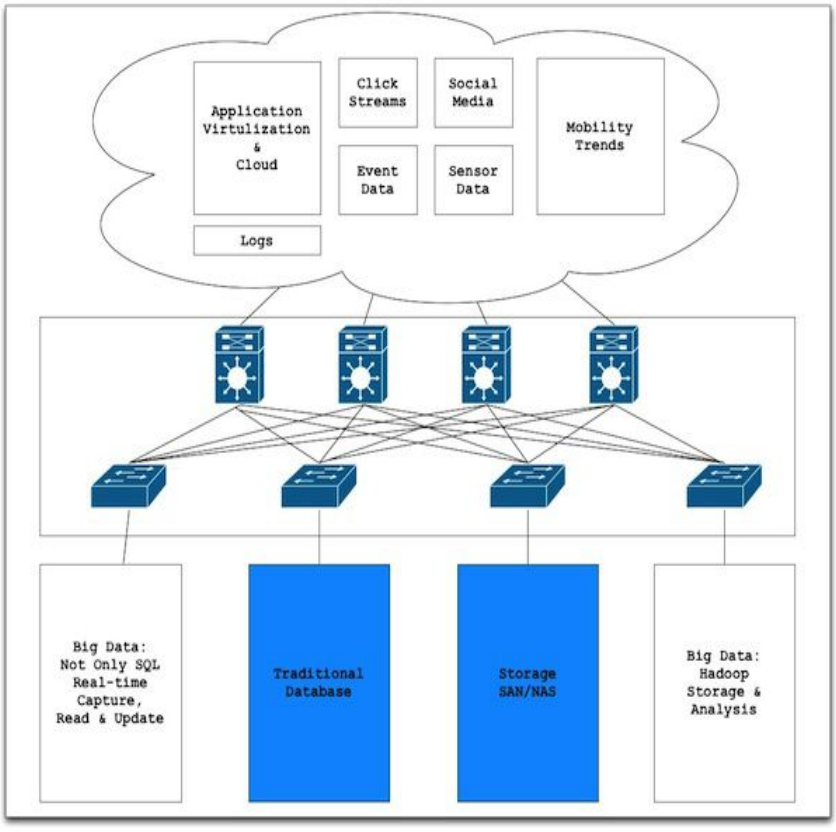

Fig 1.

HADOOP

Hadoop is designed to run on a large number of machines that don't share any memory or disks. That means you can buy a whole bunch of commodity servers, slap them in a rack, and run the Hadoop software on each one. When you want to load all of your organization's data into Hadoop, what the software does is bust that data into pieces that it then spreads across your different servers. There's no one place where you go to talk to all of your

ISSN 2349-6193
Impact Factor: 2.805

IJESMR

International Journal OF Engineering Sciences & Management Research

data; Hadoop keeps track of where the data resides. And because there are multiple copy stores, data stored on a server that goes offline or dies can be automatically replicated from a known good copy.

Hadoop Architecture, is based on HDFS, which is Hadoop distributed file system. In which data is equally (ideally) distributed on each node in the Hadoop system. When we (client) want to fetch or add or modify or delete some data from Hadoop, then Hadoop system collect the data from each node of our interest and do the meaningful actions of our interest.Hadoop [1][9][10] provides a distributed file system and a framework for the analysis and transformation of very large data sets using the MapReduce [3] paradigm. An important characteristic of Hadoop is the partitioning of data and computation across man(thousands) of hosts, and executing application computations in parallel close to their data. A Hadoop cluster scales computation capacity, storage capacity and IO bandwidth by simply adding commodity servers. Hadoop clusters at Yahoo! span 25000 servers, and store 25 petabytes of application data, with the largest cluster being 3500 servers. One hundred other organizations worldwide report using Hadoop.

Hadoop Architecture

Hadoop framework includes following four modules:

Hadoop Common: These are Java libraries and utilities required by other Hadoop modules. These libraries provides filesystem and OS level abstractions and contains the necessary Java files and scripts required to start Hadoop.

Hadoop YARN: This is a framework for job scheduling and cluster resource management.

Hadoop Distributed File System (HDFS): A distributed file system that provides high-throughput access to application data.

Hadoop MapReduce: This is YARN-based system for parallel processing of large data sets.

MAP REDUCE

Hadoop MapReduce is a software framework for distributed processing of large data sets on computing clusters. Apache Hadoop is an open-source framework that allows to store and process big data in a distributed environment across clusters of computers using simple programming models. MapReduce is the core component for data processing in Hadoop framework. In layman's term Mapreduce helps to split the input data set into a number of parts and run a program on all data parts parallel at once. The term MapReduce refers to two separate and distinct tasks. The first is the map operation, takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). The reduce operation combines those data tuples based on the key and accordingly modifies the value of the key.

Typically both the input and the output are stored in a file-system. The framework takes care of scheduling tasks, monitoring them and re-executes the failed tasks.

MapReduce Architecture.

The figure shown below illustrates the various parameters and modules that can be configured during a MapReduce operation:

ISSN 2349-6193
Impact Factor: 2.805

IJESMR
International Journal OF Engineering Sciences & Management Research

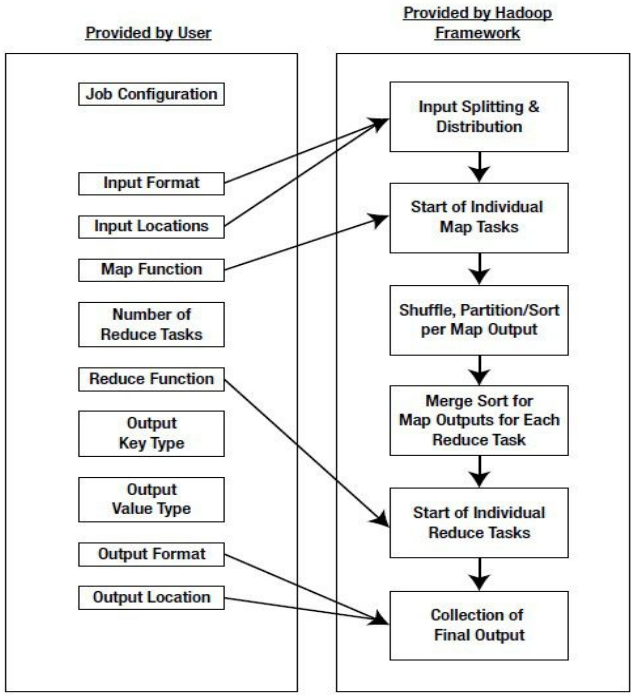

Fig. 2

The following word count example explains MapReduce method. For simplicity, let's consider a few words of a text document. We want to find the number of occurrence of each word. First the input is split to distribute the work among all the map nodes as shown in the figure. Then each word is identified and mapped to the number one. Thus the pairs also called as tuples are created. In the first mapper node three words Deer, Bear and River are passed. Thus the output of the node will be three key, value pairs with three distinct keys and value set to one. The mapping process remains the same in all the nodes. These tuples are then passed to the reduce nodes. A partitioner comes into action which carries out shuffling so that all the tuples with same key are sent to same node.

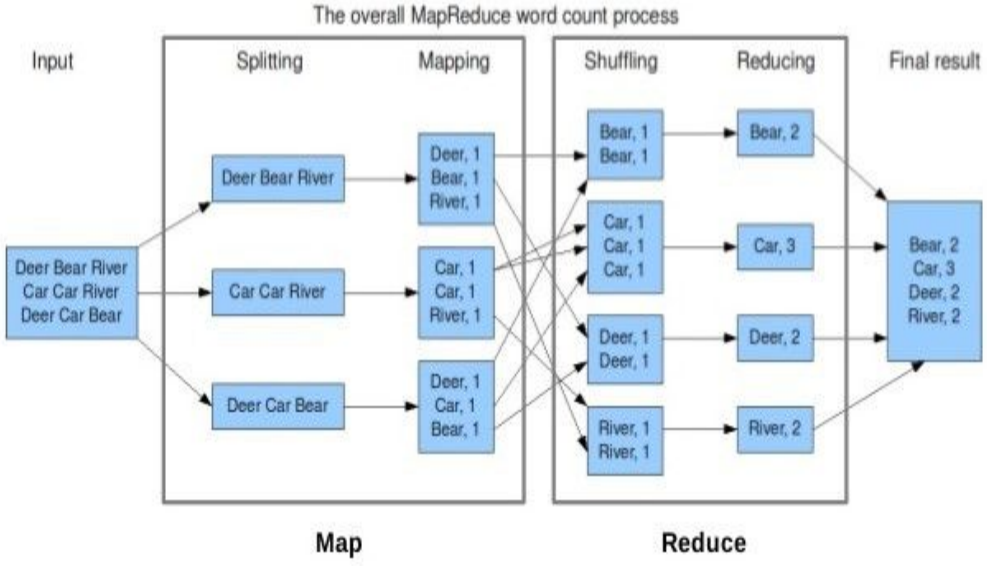

Fig. 3

ISSN 2349-6193
Impact Factor: 2.805

IJESMR

International Journal OF Engineering Sciences & Management Research

The above word count example explains MapReduce method. For simplicity, let's consider a few words of a text document. We want to find the number of occurrence of each word. First the input is split to distribute the work among all the map nodes as shown in the figure. Then each word is identified and mapped to the number one. Thus the pairs also called as tuples are created. In the first mapper node three words Deer, Bear and River are passed. Thus the output of the node will be three key, value pairs with three distinct keys and value set to one. The mapping process remains the same in all the nodes. These tuples are then passed to the reduce nodes. A partitioner comes into action which carries out shuffling so that all the tuples with same key are sent to same node.

HDFS

HDFS is structured similarly to a regular Unix filesystem except that data storage is distributed across several machines. It is not intended as a replacement to a regular filesystem, but rather as a filesystem-like layer for large distributed systems to use. It has in built mechanisms to handle machine outages, and is optimized for throughput rather than latency.

There are two and a half types of machine in a HDFS cluster:

Datanode - where HDFS actually stores the data, there are usually quite a few of these.

*Namenod*e - the 'master' machine. It controls all the meta data for the cluster. Eg - what blocks make up a file, and what datanodes those blocks are stored on.

Secondary Namenode - this is NOT a backup namenode, but is a separate service that keeps a copy of both the edit logs, and filesystem image, merging them periodically to keep the size reasonable. ◦this is soon being deprecated in favor of the backup node and the checkpoint node, but the functionality remains similar (if not the same).

HDFS Architecture

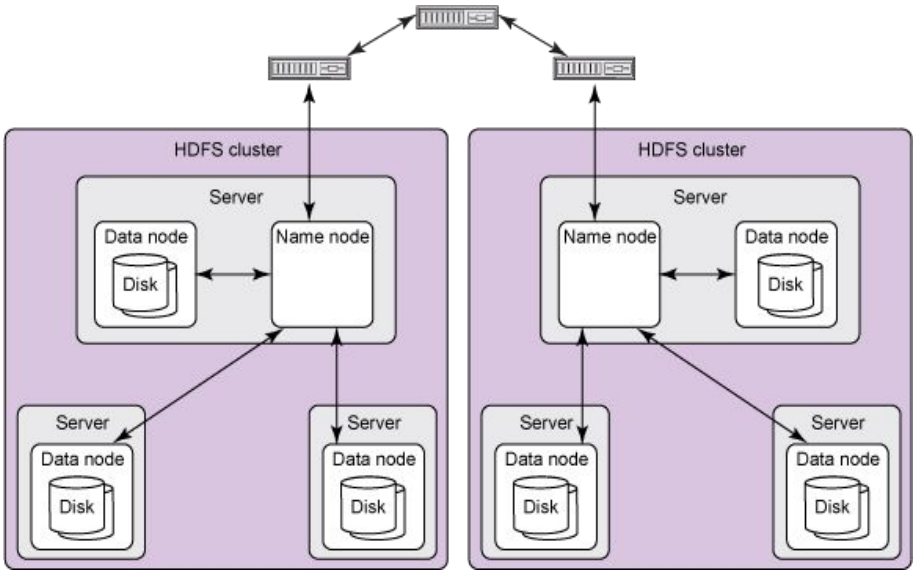

Fig. 4

Name nodes and data nodes are software components designed to run in a decoupled manner on commodity machines across heterogeneous operating systems. HDFS is built using the Java programming language; therefore, any machine that supports the Java programming language can run HDFS. A typical installation cluster has a dedicated machine that runs a name node and possibly one data node. Each of the other machines in the cluster runs one data node. Data nodes continuously loop, asking the name node for instructions. A name node can't connect directly to a data node; it simply returns values from functions invoked by a data node. Each data node maintains an open server socket so that client code or other data nodes can read or write data. The host or port for this server socket is known by the name node, which provides the information to interested

ISSN 2349-6193
Impact Factor: 2.805

IJESMR

International Journal OF Engineering Sciences & Management Research

clients or other data nodes. See the Communications protocols sidebar for more about communication between data nodes, name nodes, and clients. The name node maintains and administers changes to the file system namespace.

HDFS supports a traditional hierarchical file organization in which a user or an application can create directories and store files inside them. The file system namespace hierarchy is similar to most other existing file systems; you can create, rename, relocate, and remove files.

HDFS replicates file blocks for fault tolerance. An application can specify the number of replicas of a file at the time it is created, and this number can be changed any time after that. The name node makes all decisions concerning block replication.

HDFS uses an intelligent replica placement model for reliability and performance. Optimizing replica placement makes HDFS unique from most other distributed file systems, and is facilitated by a rack-aware replica placement policy that uses network bandwidth efficiently.

Large HDFS environments typically operate across multiple installations of computers. Communication between two data nodes in different installations is typically slower than data nodes within the same installation. Therefore, the name node attempts to optimize communications between data nodes. The name node identifies the location of data nodes by their rack IDs.

HDFS Resources

For more information about the design of HDFS, you should read through apache documentation page. In particular the streaming and data access section has some really simple and informative diagrams on how data read/writes actually happen.

CONCLUSION

Different challenges come out from the applications of Big Data Analytics, but this proposed study gives more attention on many aspects which are associated with the decomposition of Big Data System. The uniqueness of this paper is that this paper gives an overview of various techniques and highlights most of the significant findings of existing studies which is discussed briefly. This survey will be beneficial for the further progress and enhancement of Big Data Analytics in various research perspectives.

REFERENCES

1. *Apache Hadoop. http://hadoop.apache.org/.*
2. *Job Aware Scheduling Algorithm for MapReduce Framework by RadheshyamNanduri, Nitesh aheshwari, Reddy Raja, Vasudeva Varma in 3rd IEEE International Conference on Cloud Computing Technology and Science Athens, Greece.*
3. *Hadoop Tutorial: http://developer.yahoo.com/hadoop/tutorial/module1.html.*
4. *Apache Hadoop: http://www.cse.wustl.edu/~jain/cse570-13/ftp/bigdata2/.*
5. *Hadoop Tutorial: http://developer.yahoo.com/hadoop/tutorial/module1.html.*
6. *The Apache Software Foundation"http://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html.*
7. *D. Wegener, M. Mock, D. Adranale, and S. Wrobel, "Toolkit-Based High-Performance Data Mining of Large Data on MapReduce Clusters," Proc. Int'l Conf. Data Mining Workshops (ICDMW '09), pp. 296-301, 2009.*
8. *"IBM What Is Big Data: Bring Big Data to the Enterprise,"http://www-01.ibm.com/software/data/bigdata/, IBM, 2012.*
9. *T. White, Hadoop: The Definitive Guide. O'Reilly Media, Yahoo! Press,June 5, 2009.*
10. *Tom white, Hadoop Definitive Guide,Third Edition,2012.*