IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Volume-3, Issue-10 E-ISSN: 2347-2693

Big Data Platform-A Review

Sunny Kumar

Dpt. of Computer Science and Engineering, Giani Zail Singh College Bhatinda, India

www.ijcseonline.org

Received: Sep/21/2015 Revised: Oct/04/2015 Accepted: Oct /16/2015 Published: Oct /31/ 2015

***Abstract*—** Hadoop is popular distributed system used for the analysis of large amount of data. Hadoop is based on distributed computing having HDFS (Hadoop Distributed File System) &Map Reduce programming paradigm. Hadoop is highly fault-tolerant due to its imitation of data transversely on multiple nodes and can be set out on low cost hardware. The file system – HDFS—written in JAVA and designed for heterogeneous hardware and software. Hadoop is very much appropriate for high volume of data & where data format is different like semi structured, unstructured. Hadoop also make available the high speed admittance to the data of the application which we want to use. Hadoop architecture is cluster based (cluster consists of racks), which is consist of nodes (data note, name node), physically separate to each other, in idyllic circumstances. In Hadoop a program known as map-reduce is used to collect data according to query. As Hadoop is used for massive amount of data therefore scheduling and way of containing data in Hadoop must be efficient for better presentation. With this feature of Hadoop the traditional system is replacing with Hadoop. The research objective is to study and explore various scheduling techniques, which are used to increase performance in Hadoop. This paper include the idea of working of Hadoop, its internal details and why Hadoop is better than the Traditional system.

***Keywords*—** Hadoop, HDFS, Name node, Data node. Map Reduce, Data locality, Job Tracker, Task Tracker

I. INTRODUCTION

Hadoop is an open source venture of apache foundation written in JAVA. Hadoop uses google file classification and google Map Reduce. It is optimized to knob massive magnitude of data which could be structured, unstructured or semi- structured using commodity hardware with great performance. The superior fixation about HADOOP is its reproduction of data across multiple nodes, so that if any node goes down data is processed on one of the replicated nodes. It is a kind of batch operation handling of massive amount of data. The rejoinder time is not instantaneous in Hadoop and this is one of one of biggest reason why Hadoop is not suitable for

OLAP &OLTP.

One of the key mechanisms of Hadoop is the redundancy built into the milieu. Not only is the data redundantly stored in multiple places across the cluster, but the programming model is such that failures are expected and are determined robotically by running segments of the program on diverse servers in the cluster. In the Traditional system for easy availability of data we have to use extra hardware resources and the management of increasing day by day data is becoming difficult with it. So, Hadoop is introduced to overcome this difficulty. The ambition of Hadoop is to use frequently and universally available servers in a very large cluster, where each server has a set of reasonably priced internal disk drives. For higher performance, Map Reduce tries to assign workloads to these servers where the data to be progressed is accumulated. This is known as ***data locality*.**

Corresponding Author: *Sunny Kumar, sunnykumar1018@gmail.com*
Department of Computer Science and Engineering, Giani Zail Singh Bhatinda, India

II. WHY WE NEED HADOOP

The advancement of new technology has emerged a very adverse effect on generating a large amount of data. Each and every sector are suffering from the problem of storing and analysis a huge amount of data.

1. Let's reflect on to a state of affair where you have 2 GB of data and you want to process it. Very easy task data stored in relational database management system on your desktop computer and computer has no problem to handle this data. Now think about you have 10 GB Data and then 100 GB data and you start to accomplish the confines of your computer. At this split now you think about empowering money in large configuration computer and now you are ok. After few months your data grows 100GB to 20 TB and then 150 TB and moreover data format is of type well thought-out, formless & semi structured and your executive wants quick rejoinder from all this data as

soon as possible. Now graving state of affair occur what should we do now? Solution of this problem is HADOOP. HADOOP is used to handle this gigantic quantity of data with the commodity hardware approach.

Populace and organizations have attempted to embark upon this dilemma from many poles apart angles. Of course, the perspective that is currently leading the pack in terms of popularity for massive data analysis is an open source project called *Hadoop* that is shipped as part of the IBM Info Sphere Big Insights (Big Insights) platform. [1]

2. 100 files

50 (Processed) 100(More Files)

50(Processed) 100(More Files)

50(Processed)

At first you have 100 files which have a size of 100 GB and you are easily processing 50 files at a time and rest you process later. Earlier situation is that you have small data and you can easily process according to your need. But now situation is different you have more 100 files that have same size 100 GB and now you process only 50 same as previous case. Now you have 100 files that are not yet processed. Again data increases and 100 more files come. So situation is totally different data increases at a fast rate but your processing capacity is same and your output is not good .So now day's data is increasing day by day and processing speed is also increased in some different way so our response will be good. For that same work is divided into multiple tasks that can run parrallelly so that less time should be taken. So best solution for big data is Hadoop.

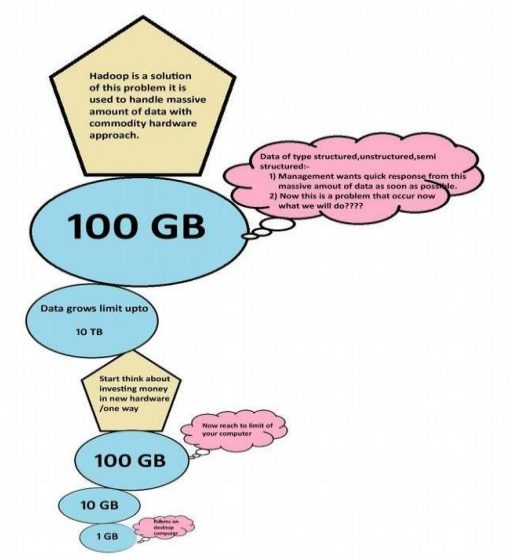

Figure 1. Diagrammatic view-why hadoop for big data

III. FUNCTIONING OF HADOOP

A. *Hadoop distributed file system*

It is designed especially for storing large number of data set in a cluster using commodity hardware with streaming access pattern. It runs on top of existing file coordination on each node in a HADOOP cluster. A Hadoop block is a file on the underlying file system. Since the underlying filesystem stores files as blocks, one Hadoop block may consist of many blocks in the underlying file system. Blocks are large. They default to 64 megabytes each and most systems run with block sizes of 128 megabytes or larger. Finally, blocks fit well with replication, which allows HDFS to be fault tolerant and obtainable on commodity hardware.

As shown in the figure: Each block is replicated to multiple nodes.

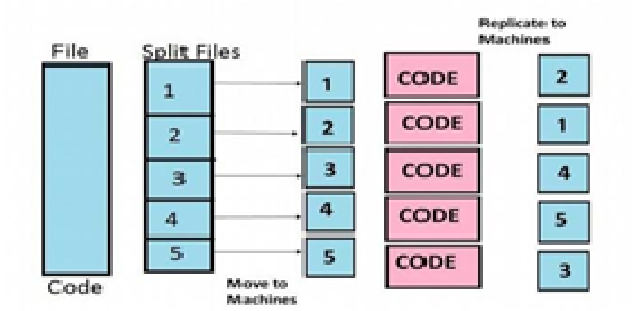

Figure 2: How HDFS works

B. *MapReduce Engine*

Map Reduce the heart of Hadoop. It is this indoctrination archetype that consent for massive scalability across hundreds or thousands of servers in a Hadoop cluster.

MapReduce is an encoding paradigm, in which work is broken down into mapper and reducer tasks to manipulate data that is stored across a cluster of servers for massive parallelism.

A MapReduce program consists of two types of transformations that can be applied to data any number of times - a map renovation and a reduce transformation. A MapReduce job is an implementing MapReduce program that is divided into map tasks that run in analogous with each other and trim down tasks that run in parallel with each other.

The term *MapReduce* actually refers to two separate and divergent errands that Hadoop programs execute. The first is the map job, which takes a set of data and converts it into another set of data, where individual fundamentals are conked out down into *tuples* (key/value pairs). The trimmed down job takes the output from a map as input and coalesces those data tuples into a minor set of tuples. As the progression of the name MapReduce implies, the reduce job is always performed after the map job. [6]

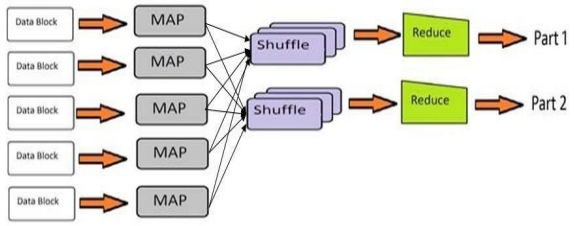

Figure 3. The flow of data in a simple MapReduce job

IV. MEDIA OF HADOOP

For HDFS nodes we have the Name Node, and the Data Nodes. For MapReduce version 1 nodes we have the Job Tracker and the Task Tracker nodes.

HDFS services:

1. Name Node (Master Node)
2. Secondary Name Node
3. Job Tracker (Master Node)
4. Task Tracker (Slave Node)
5. Data Node (Slave Node)

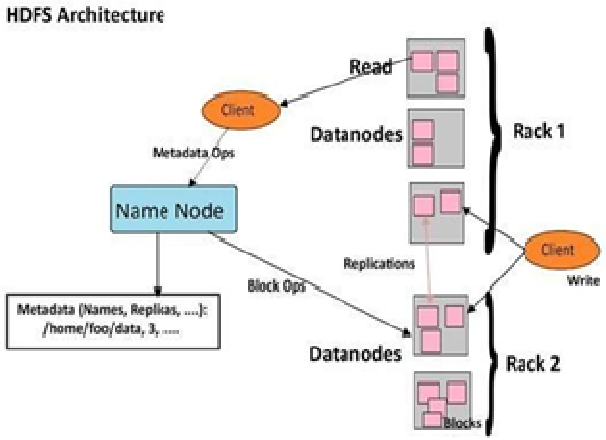

Figure 4. Architecture of hadoop file system

A client is shown as communicating with a Job Tracker. It can also correspond with the Name Node and with any Data Node. There is only one Name Node in the cluster. The metadata for a file is stored at the NameNode. The NameNode is also responsible for the filesystem namespace. A typical HDFS cluster has many DataNodes. DataNodes store the blocks of data and blocks from different files can be stored on the same DataNode. When a client requests a file, the client finds out from the NameNode which DataNodes stored the blocks that make up that file and the client directly reads the blocks from the individual DataNodes. Each Data Node also reports to the NameNode sporadically with the list of blocks it stores. DataNodes do not necessitate classy and unrestrained enterprise hardware or imitation at the hardware layer. The DataNodes are premeditated to run on commodity hardware and imitation is provided at the software layer.

A Job Tracker node manages MapReduce V1 jobs. There is only one of these on the cluster. It receives jobs submitted by clients. It schedules the Map tasks and Reduce tasks on the appropriate Task Trackers that is where the data resides, in a rack-aware comportment and it monitors for any failing tasks that need to be rescheduled on a different TaskTracker. To accomplish the parallelism for your map and reduce tasks, there are many TaskTrackers in a Hadoop cluster. Each TaskTracker spawns Java Virtual Machines to run your map or reduce task. It corresponds with the JobTracker and reads blocks from DataNodes.

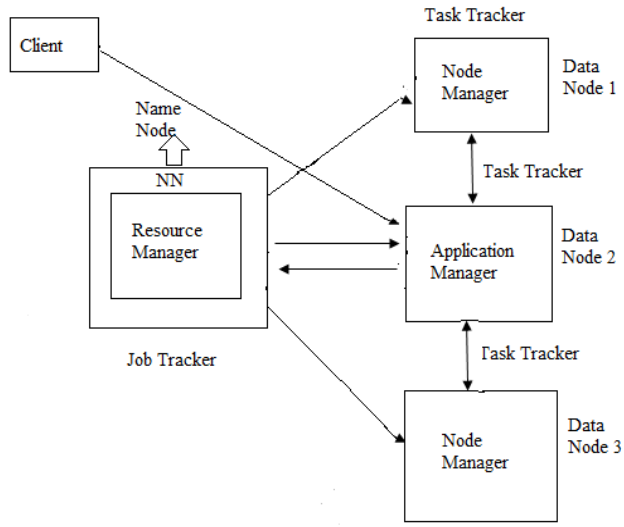

Figure 5 Job processing in hadoop

V. TRADITIONAL SYSTEM VS HADOOP

Hadoop is a quickly budding ecosystem of components based on Google's MapReduce algorithm and file system work for implementing MapReduce algorithms in a scalable fashion and distributed computing on commodity hardware. Hadoop enables users to store and process large volumes of data and analyze it in ways not previously possible with SQL-based approaches or less scalable solutions.

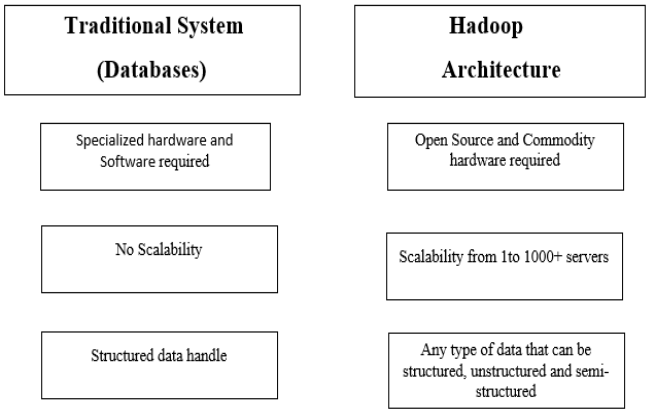

Figure6 Difference between traditional system and Hadoop architecture

1. **HDFS:** The main advantage of using hdfs in Hadoop is that it can run on the top of any file system. HDFS comes into picture and take care of all the issues related to file system.

2. **BLOCK SIZE: T**he default block size of any hdd in any operating system is 4 kb. When HADOOP is installed Its HDFS component has block size 64 Mb by default. And it may be extended to 124 MB.

3. **MEMORY UTILIZATION:** Memory utilization is more and better in HDFS rather than in any other file system. Suppose in any other file system the default size is 4kb and we are using a 2 kb at a time, rest 2kb space of block is wasted as no other block use it. But the scenario is totally different in HDFS. Suppose we have 64 MB block size and a request of 32 MB block come then rest of 32 MB space will be utilized by some other block.

4. **REPLICATION:** The main advantage of Hadoop is that it provides replication of data between its data node. It replicates data between its data nodes. The architecture of Hadoop provide replication factor 3 by default.

5. **FAULT TOLERANCE:** Because of replication of data it provides high availability of data between its data nodes .If at a time anyone data node gets slow then other data node can provide services because of replication of data.

6. **VARIETY OF DATA IN HADOOP:** As traditional system can process most of structured data. But Hadoop architecture support varieties of data like structured, unstructured and semi structured. In today's world most of data is in unstructured format.

VI. CONCLUSION

As we know that data is greater than ever day by day in the configure of structured, semi structured & unstructured.to handle this massive quantity of data HADOOP expertise is used. We have confer various workings of Hadoop in this paper that show HADOOP is a fault tolerance system that is highly reliable as compared to other technology system. The main shortcoming of HDFS is that it enclose only one NameNode which handle all metadata operations. But we can overcome this drawback by introducing multiple NameNodes. We also discuss that Big Data is not just Hadoop but It is supplementary than Hadoop .Hadoop is just to administer and stockpile large amount of data & to handle & supervise streaming data, analyze structure and control data various other components also used.

REFERENCES

[1] Transl. J. Magn. Japan, [Digests 9th Annual Conf. Magnetics Japan, Vol. 2, pp. 740-741, August 1987 pp. 301, 1982].

[2] Chris Eaton and Tom Deutsch, Understanding Big Data-Analytics for Enterprise Class Hadoop and Streaming Data.

[3] Arun C. Murthy and Vinod Kumar Vavilapalli, Apache Hadoop YARN-Moving beyond MapReduce and Batch Processing with Apache Hadoop 2.

[4] http://www.bigdatauniversity.com/web/media/player.php?file=BD001V212EN/Videos/Unit_1_What_is_Hadoop_Part1.mp4&caption=files.db2university.com/BD001V212EN/Videos/EN/Unit_1_What_i s_Hadoop_Part1.srt

[5] https://www.youtube.com/watch?v=DLutRT6K2rM

[6] Figure 2. The flow of data in a simple MapReduce job pp.62 Chris Eaton and Tom Deutsch, Understanding Big Data- Analytics for Enterprise Class Hadoop and Streaming Data.