

# A STUDY ON HADOOP ARCHITECTURE FOR BIG DATA ANALYTICS

**Dr. Ananthi Sheshasayee<sup>1</sup>, J V N Lakshmi<sup>2</sup>**

<sup>1</sup>Head Department of Computer Science & Research, Quaid-E-Millath Govt College for Women, Chennai, (India)

<sup>2</sup>Research Department of Computer Science, SCSVMV University, Kanchipuram, (India)

## ABSTRACT

*Big Data Analytics is now a key ingredient for success in many business organizations, scientific and engineering disciplines and government endeavors. The data management has become a challenging issue for network centric applications which need to process large amount of data sets. We require advanced tools to analyse these data sets. Parallel data bases are well suited to perform analytics. Map Reduce paradigm parallelize massive data sets using clusters or grids. Hadoop is an open source project which uses Google's Map Reduce. It is optimized to handle massive quantities of data with available execution engine, pluggable distributed storage engines and a range of procedural interfaces using commodity hardware. This massive parallel processing uses a distributed file system known as HDFS. This paper discusses the architecture of HDFS.*

**Keywords: Analytics, Hadoop, HDFS, Map Reduce.**

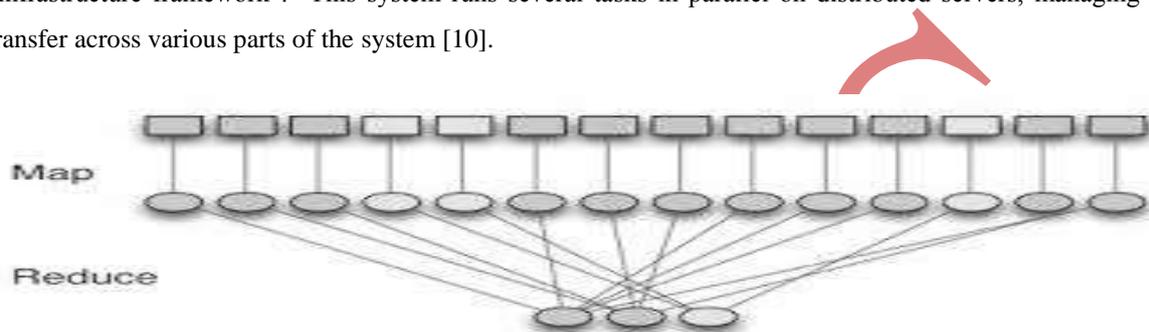
## I. INTRODUCTION

Data Analytics is a process of inspecting, cleaning, transforming, and modeling data with the goal of discovering useful information, suggesting conclusions, and supporting decision making. Data analysis has multiple facets and approaches, encompassing diverse techniques under a variety of names, in different business, science, and social science domains. Big Data is a particular data analysis technique that focuses on analytics of massive data set, which emerges from diverse field of data intensive informatics. Most of the data is unruly stuff like words, images and video on web which is unstructured [11]. Such text lacks a structure which makes the data impossible to analyze using traditional database techniques [4]. Diverse fields of technological domains, Government agencies, many business organizations and sociological sites produce massive data. Big Data usually includes data sets with the sizes beyond the ability of common software tools to analyse, manage and process the data [3]. Big data tools are developed to handle huge quantities of data. Map Reduce is a generic programming model for processing large data sets [1]. A Map Reduce program comprises two functional procedures Map() and Reduce(). Hadoop is an open source project; it has two major components, the distributed file system component and the Map Reduce component. The emphasis on a distributed file system called Hadoop Distributed File System or HDFS is used for parallel processing [6]. Section 2 describes the basic programming model Map Reduce interface tailored towards our cluster – based computing environment. Section 3 describes Hadoop architecture. In Section 4, we discuss about HDFS for parallel processing. Section 5 describes Map Reduce with HDFS. Section 6 concludes the paper.

## II. MAP – REDUCE PARADIGM

Map Reduce is a data flow paradigm for such applications [6]. It's simple, explicit data flow programming model, favored over the traditional high level data base approaches. Map Reduce paradigm parallelize huge data sets using clusters or grids. A Map Reduce program comprises of two functions, a *Map( )* Procedure and a *Reduce( )* Procedure.

A *Map( )* procedure performs filtering and sorting by managing the “infrastructure”. Whereas *Reduce( )* procedure summarizes the operations designing the “framework”. Therefore, the system is referred as “infrastructure framework”. This system runs several tasks in parallel on distributed servers, managing data transfer across various parts of the system [10].



**Fig 1: Map Reduce Framework**

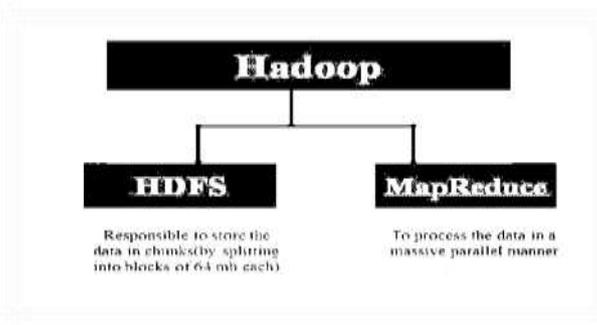
**Source: [14]**

This figure 1 depicts the data from various sources is taken as input and the is mapped by calling a *Map()*. Then mapped data gives an output that data is reduced by calling the *Reduce()* procedure. This results the output data. This practice of Map Reduce paradigm slows down data analytics, makes processing of data difficult and impede automated optimization [7]. So alternative to store and process extremely large data sets, development of a popular open source implementation Hadoop Map Reduce is derived [6].

## III. HADOOP (HIGH AVAILABILITY DISTRIBUTED OBJECT ORIENTED PLATFORM)

Hadoop uses Google's MapReduce and Google File System technologies as its foundation [11]. It is optimized to handle massive quantities of data which could be structured, unstructured or semi-structured, using commodity hardware, that is, relatively inexpensive computers [4]. Hadoop replicates its data across different computers, so that if one goes down, the data are processed on one of the replicated computers. Hadoop is used for Big Data. It complements OnLine Transaction Processing and OnLine Analytical Processing. Yahoo is also the largest contributor to the Hadoop open source project. They can integrate analytic solutions to the mix to derive valuable information that can combine structured legacy data with new unstructured data [8]. Two major components of Hadoop, the distributed file system component and the MapReduce component, with an emphasis on a distributed filesystem called Hadoop Distributed File System or HDFS an important feature of Hadoop called "rack awareness" or "network topology awareness".

A node is simply a computer, typically non-enterprise, commodity hardware that contains data. A rack is a collection of 30 or 40 nodes that are physically stored close together and are all connected to the same network switch. Network bandwidth between any two nodes in rack is greater than bandwidth between two nodes on different racks. A Hadoop Cluster (or just 'cluster' from now on) is a collection of racks.



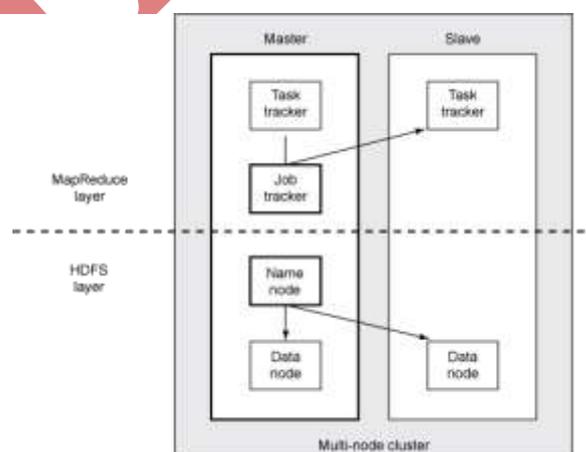
**Fig 2: Hadoop Components**

**Source: [14]**

Hadoop is designed for streaming or sequential data access rather than random access. Sequential data access means fewer seeks, since it only seeks to the beginning of each block and begins reading sequentially from there. Hadoop uses blocks to store a file or parts of a file [15]. A Hadoop block is a file on the underlying file system. Since the underlying file system stores files as blocks, one Hadoop block may consist of many blocks in the underlying file system Blocks[2]. Blocks have several advantages: First, they are fixed in size. This makes it easy to calculate how many can fit on a disk. Second, by being made up of blocks that can be spread over multiple nodes, a file can be larger than any single disk in the cluster.

#### **IV. HDFS - (HADOOP DISTRIBUTED FILE SYSTEM)**

Types of nodes in Hadoop are classified as HDFS or MapReduce nodes. For each HDFS node there is a NameNode, and a DataNode. For MapReduce nodes a JobTracker and a TaskTracker nodes exists. There are other HDFS nodes such as the Secondary NameNode. A typical HDFS cluster has many DataNodes. They store the blocks of data and when a client requests a file, it finds out from the NameNode which DataNodes store 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 periodically with the list of blocks it stores. DataNodes do not require expensive enterprise hardware and replication is provided at the software layer. The DataNodes are designed to run on commodity hardware and replication is provided at the software layer.



**Fig 3: HDFS Architecture**

**Source: [12]**

This diagram shows some of the communication paths between the different types of nodes on the system. A client can communicate with a JobTracker. It can also communicate with the NameNode and with any Data Node. There is only one NameNode in the cluster. While the data that makes up a file is stored in blocks at the data nodes, the metadata for a file is stored at the NameNode. The NameNode is also responsible for the file system namespace.

A JobTracker node manages MapReduce jobs one on a cluster. Jobs received from client are scheduled to a TaskTracker to monitor for any failing tasks that need to be rescheduled on a different TaskTracker. To achieve the parallelism for your map and reduce tasks, there are many TaskTracker in a Hadoop cluster. Hadoop uses rack awareness topology of the network which allows it to optimize where it sends the computations to be applied to the data. It operates on maximizes the bandwidth available for reading the data [13].

## V. MAP REDUCE ON HDFS

The process of running a MapReduce job on Hadoop consists of 8 major steps. The first step is the MapReduce program tells the JobClient to run a MapReduce job. This sends a message to the JobTracker which produces a unique ID for the job. The JobClient copies job resources, such as a jar file containing a Java code you have written to implement the map or the reduce task, to the shared file system, usually HDFS. Once the resources are in HDFS, the JobClient can tell the JobTracker to start the job. The JobTracker does its own initialization for the job. It calculates how to split the data so that it can send each "split" to a different mapper process to maximize throughput. It retrieves these "input splits" from the distributed file system. The TaskTrackers are continually sending heartbeat messages to the JobTracker. Now that the JobTracker has work for them, it will return a map task or a reduce task as a response to the heartbeat. The TaskTrackers obtains the code to execute from the shared file system [9].

## VI. CONCLUSION

Hadoop MapReduce is an implementation of map and reduce that is parallel, distributed, fault-tolerant and network topology-aware. This lets system run efficiently using map and reduce operations over large amounts of data on HDFS. The Hadoop system (HDFS) has advantages over a DBMS in some circumstances. "As a file system, HDFS can handle any file or document type containing data that ranges from structured data (relational or hierarchical) to unstructured data (such as human language text)," says Russom. When HDFS and MapReduce are combined, Hadoop easily parses and indexes the full range of data types. Furthermore, as a distributed system, HDFS scales well and has a certain amount of fault tolerance based on data replication, even when deployed atop commodity hardware. For these reasons, HDFS and MapReduce (whether from the open source Apache Software Foundation or elsewhere) can complement existing traditional DW systems that focus on structured, relational data. Furthermore, the MapReduce component of Hadoop brings advanced analytic processing to the data. This is the reverse of older practices where large quantities of transformed data were brought to an analytic tool, especially those based on data mining or statistical analysis. As big data gets bigger, it's just not practical (from both a time and cost perspective) to move and process that much data[5]. The HDFS is gaining popularity to store data on a massively parallel collection of inexpensive commodity hardware. Typically these approaches involve MapReduce, a programming model for processing and generating large data sets that is automatically paralleled and executed on a large cluster of commodity machines.

## REFERENCES

### Conference Papers:

- [1] For Big Data Analytics There's No Such Thing as Too Big [The Compelling Economics and Technology of Big Data Computing] March 2012 by: 4syth.com Emerging big data thought leaders.
- [2] StarFish - A self tuning system for Big Data Analytics, Herodotos, Harald Lim, GangmLuo CIDR '11 USA
- [3] L Huston, R Wickremesinghe, M SatyaNarayana – Storage Architecture for early discard in interactive search – FAST conference proceedings 2004
- [4] C.Ronnie et al. SCOPE: Easy and Efficient Parallel Processing of Massive Data Sets. Proc. VLDB 2008

### Journal Papers:

- [5] A. Pavlo et. al. A Comparison of Approaches to Large-Scale Data Analysis. Proc. ACM SIGMOD, 2009.
- [6] Mapreduce: A flexible data processing tool, ACM Communications, Volume 53, pp 72-77, 2010.
- [7] HIVE – A warehousing solution over a Map – Reduce framework Ashish Thusoo, Joy deep Sen Sharma VLDB '09
- [8] Abouzeid, A., Bajda-Pawlikowski, K., Abadi, D.J... Rasin, A., and Silberschatz, A. HadoopDB: An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical Workloads. *PVLDB 2(1)*, 2009.
- [9] Building a HighLevel Dataflow System on top of MapReduce: The Pig Experience 2009 Alan F. Gates, Olga Natkovich, Shubham Chopra, Pradeep Kamath, Shravan M. Narayanamurthy
- [10] Dean, Jeff and Ghemawat, Sanjay. MapReduce: Simplified Data Processing on Large Clusters.2004
- [11] Luiz A. Barroso, Jeffrey Dean, and Urs Holzle. " Web search for a planet: The Google cluster architecture. IEEE Micro, 23(2):22–28, April 2003.

### WebSites:

- [12] Hadoop: Open-source implementation of MapReduce. <http://hadoop.apache.org>
- [13] BigDataUniversity.com
- [14] Google.com

### Books:

- [15] Hadoop – A definitive Guide