# A REVIEW: MAPREDUCE AND SPARK FOR BIG DATA ANALYTICS

**Meenakshi Sharma[1], Vaishali Chauhan[2], Keshav Kishore[3]**

*[1,2]Students of Master of Technology, A P Goyal Shimla University, (India)*

*[3]Head of department, Department of Computer Science and Engineering,*

*Department of Computer Science and Engineering, A P Goyal Shimla University, (India)*

## ABSTRACT

*In this paper we discuss the various challenges of Big Data and problem arises due to continuous explosion of data resulting from the likes of social media and other online sources to gain access to deeper analysis of their data. This paper discusses two of the comparison of Hadoop Map Reduce and the recently introduced Apache Spark – both of which provide a processing model for analyzing big data. Although both of these options are based on the concept of Big Data, their performance varies significantly based on the use case under implementation. Data growing at very high speed and is having very large volume. Presently, to assemble the large volume of dataset at lesser cost, storage technology and data collection has made it possible for any organization.*

***Keywords: Hadoop, HDFS , MAPREDUCE, SPARK , Big data analytics.***

## I. INTRODUCTION

Big data is the name used every where now a days in distributed paradigm on web. As the name shows it is the collection of sets of very huge amount of data in terabytes, pet bytes etc. associated with systems as well as algorithms used to analyze this massive data. [5] Big Data is a collection of data sets so large and complex that it becomes difficult to process using on-hand database management tools.**[1]** For instance, an International Data Corporation **(**IDC) report predicts that, from 2005 to 2020, the global data volume will grow by a factor of 300, from 130 Exabyte to 40,000 Exabyte, representing a double growth every two year.[2] IBM indicates that 2.5 Exabyte data is formed every day that is extremely tough to investigate. The estimation about the generated data is that till 2003 it was represented about 5 EB of data, then until 2012 is 2.7 ZB of data and till 2015 it is expected to increase 3 times.[3] The need of Big Data comes from the big Companies like yahoo, Google, facebook etc for the purpose of analysis of big amount of data which is in unstructured form. Traditional data management and analysis systems are based on the relational database management system (RDBMS). It is evident that the traditional RDBMS could not handle the huge volume and heterogeneity of Big Data. For solutions of permanent storage and management of large-scale disordered datasets, hadoop distributed file systems and NoSQL (Not Only SQL) Databases are good choices [7]

Big-data system faces a series of technical challenges, including: First, due to the variety of disparate data sources and the sheer volume, it is difficult to collect and integrate data with scalability from distributed locations. For instance, more than 175 million tweets containing text, image, video, social relationship are generated by millions of accounts distributed globally [4]. Second, big data systems need to store and manage the gathered massive and heterogeneous datasets, while provide function and performance guarantee, in terms of fast retrieval, scalability, and privacy protection. For example, Facebook needs to store, access, and analyze over 30 petabytes of user generate data .

Third, big data analytics must effectively mine massive datasets at different levels in real-time or near real-time - including modeling, visualization, prediction, and optimization - such that inherent promises can be revealed to improve decision making and acquire further advantages.[5]

Big Data analysis, including Google's MapReduce, Yahoo's PNUTS , Microsoft's SCOPE , Twitter's Storm and spark, LinkedIn's Kafka and Walmart. Also, several companies, including Facebook, both use and have contributed to Apache Hadoop (an open-source implementation of MapReduce) and its ecosystem.[6]

## II. CHARACTERISTICS OF BIG DATA

The characteristics of the big data depends on the three factors which includes Data Velocity, Data Volume and Data Variety**.** Big Data is not just about the size of data but also includes data variety and data velocity. these are the five V’s of the Big data.[1]

**2.1 Volume**: Big data denotes its massive character, i.e. a huge amount of information involved. Data is ever-growing day by day of all types ever Kilo Byte, Mega Byte, Peta Byte, Yotta Byte, Zetta Byte, Tera Byte of information. The data results into massive files. Excessive volume of information is main issues of storage. This main issue is resolved by reducing storage value**.** Data volumes are expected to grow more than 50 times by 2020.

**2.2 Variety:** Data sources (even in the same field or in distinct) are extremely heterogeneous. The files comes in various formats and of any type, it may be unstructured or structured such as text, audio, log files, videos and more. The varieties are endless, and the data enters the network without having been quantified or qualified in any way.[2]

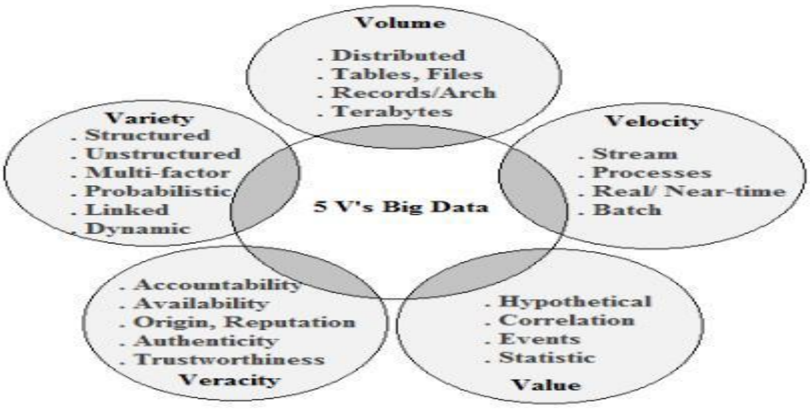

**Fig-1 Five V’s of Big Data**

**2.3 Velocity:** The data comes at high speed. Sometimes one minute is too late so big data is time sensitive. Most organisations data velocity is main challenge. The credit card transactions and social media messages done in millisecond and data generated by this putting in to databases.

**2.4 Value:** Which addresses the requirement for valuation of enterprise data? It is a most important V in big data. Value is main buzz for big data because it is important for IT infrastructure system, businesses to store large amount of values in database.

**2.5 Veracity:** The increase in the range of values typical of a large data set. When we dealing with high volume, velocity and variety of data, the all of data are not going 100% correct, there will be dirty data. Big data and analytics technologies work with these types of data.

## III. BIG DATA ANALYTICS

Big data analytics is the process of using analysis algorithms running on powerful supporting platforms to uncover potentials concealed in big data, such as hidden patterns or unknown correlations. According to the processing time requirement, big data analytics can be categorized into two alternative paradigms.[13]

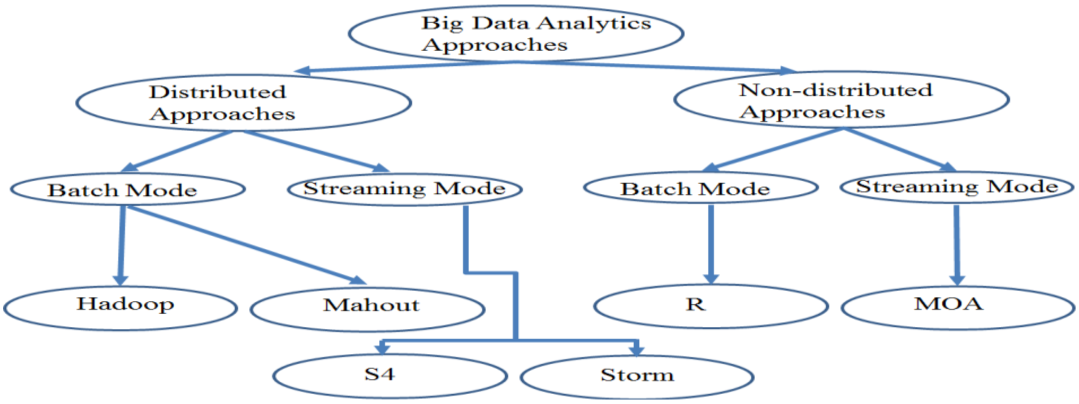

Fig-2 Taxonomy of Big data analytics

**3.1 Batch Processing:** In the batch-processing paradigm, data are first stored and then analyzed. MapReduce has become the dominant batch-processing model. The core idea of MapReduce is that data are first divided into small chunks. Next, these chunks are processed in parallel and in a distributed manner to generate intermediate results. The final result is derived by aggregating all the intermediate results.[12 This model schedules computation resources close to data location, which avoids the communication overhead of data transmission. The MapReduce model is simple and widely applied in bioinformatics, web mining, and machine learning

**3.2 Streaming Processing:** The start point for the streaming processing paradigm is the assumption that the potential value of data depends on data freshness. Thus, the streaming processing paradigm analyzes data as soon as possible to derive its results. In this paradigm, data arrives in a stream. In its continuous arrival, because the stream is fast and carries enormous volume, only a small portion of the stream is stored in limited memory. One or few passes over the stream are made to find approximation results. Streaming processing theory and technology have been studied for decades. Representative open source systems include Spark, Storm .[9] The streaming processing paradigm is used for online applications, commonly at the second, or even millisecond, level.

## IV. BIG DATA ANALYTICS TOOLS

In This we have two mode of data processing. One is batch processing mode which uses the Mapreduce framework of Hadoop and second is streaming mode which is real time processing uses the Spark for data processing.

### 4.1 Hadoop

Hadoop is a Programming framework used to support the processing of large data sets in a distributed computing environment.[3] Hadoop was developed by Google's Mapreduce that is a software framework where an application break down into various parts. The Current Apache Hadoop ecosystem consists of the Hadoop Kernel, Mapreduce, HDFS and numbers of various components like Apache Hive, HBase and Zookeeper, pig, Oozie, spark. In this there are lots of components of Hadoop but I am going to explain HDFS and Spark as storage and processing purpose. In this HDFS is used for the storage of large data set and processing is used by Mapreduce and Spark. There are some shortcoming in Mapreduce so that spark is a new component that overcome the Mapreduce issues.

### 4.1.1 Hadoop Distributed File System – HDFS

Hadoop includes a fault-tolerant storage system called the Hadoop Distributed File System, or HDFS. Filesystem that manage the storage across a number of machine are called distributed Filesystem. It is applied when the amount of data is too much for a single machine.

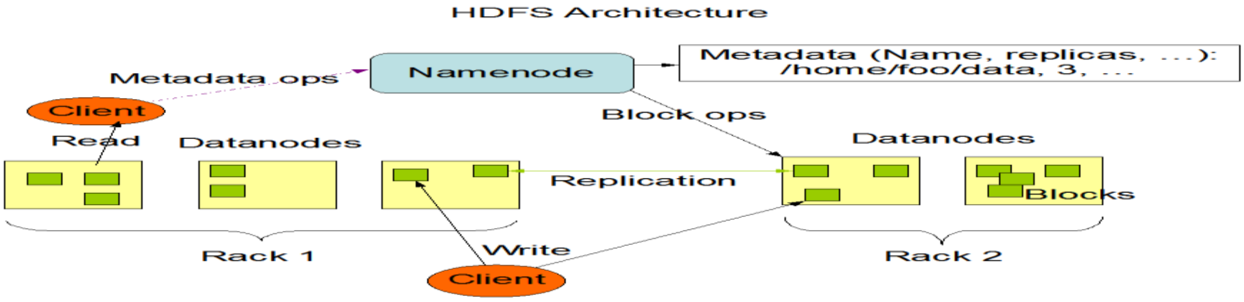

Fig- 3: HDFS Architecture

The role of HDFS is to split data into smaller blocks and distribute it throughout the cluster. The name node stores the metadata for the Name Node .Name Nodes keeps track of the state of the Data Nodes.. Name Node is also responsible for the file system operations etc.[8]

### 4.1.2 Mapreduce

MapReduce is a programming framework for distributed computing which was created by Google using the divide and conquer method to break down complex big data problems into small units of work and process them in parallel. MapReduce can be divided into two stages:[11]

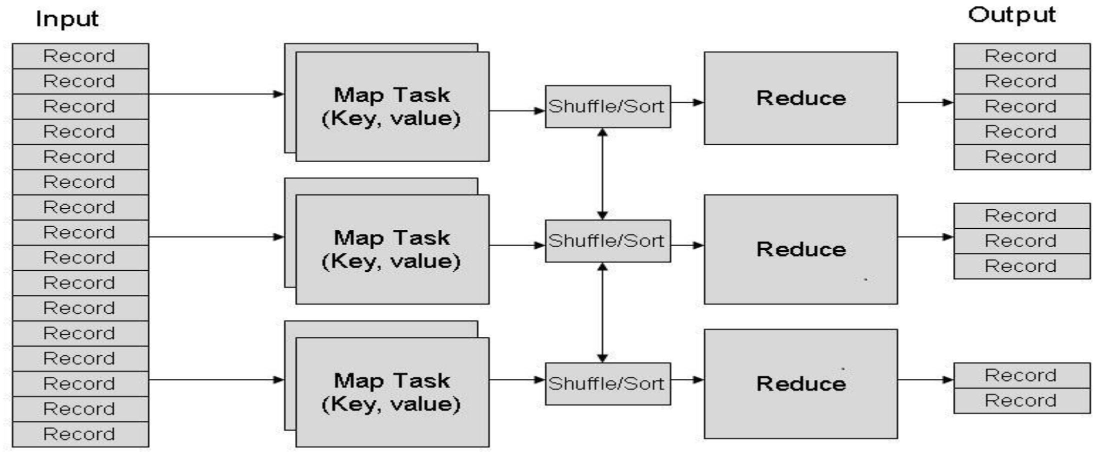

Fig-4 Map Reduce Architecture and Working

The Map Reduce programming model consists of two functions, map () and reduce (). Users can implement their own processing logic by specifying a customized map() and reduce() function. The map () function takes an input key/value pair and produces a list of intermediate key/value pairs. The Map Reduce runtime system groups together all intermediate pairs based on the intermediate keys and passes them to reduce() function for producing the final results.[9]

Map Reduce can be divided into two steps:

**a) Mapper Step**

Mapper maps input key/value pairs to a set of intermediate key/value pairs. Maps are the individual tasks that transform input records into intermediate records. The transformed intermediate records do not need to be of the same type as the input records. A given input pair may map to zero or many output pairs

**b) Reducer Step:** Reducer has **3** primary phases: shuffle, sort and reduce.

**1. Shuffle:** Shuffling is a phase on intermediate data to combine all values into a collection associated to same key. After this there will be no duplicate key in intermediate data.

**2. Sort:** The set of intermediate keys on a single node is automatically sorted by Hadoop before they are presented to the Reducer. Sorting is done because of Box Classes. The shuffle and sort phases occur simultaneously; while map-outputs are being fetched they are merged.

**3. Reduce:** Shuffled and sorted output data of mapper is provided to Reducer. In this phase the reduce(Writable Comparable, Iterator, Output Collector, Reporter) method is called for each <key, (list of values)> pair in the grouped inputs.

### 4.3. APACHE SPARK

Apache Spark is a lightning-fast cluster computing technology, designed for fast computation. Spark is a next generation paradigm for big data processing developed by researchers at the University of California at Berkeley. It is an alternative to Hadoop which is designed to overcome the disk I/O limitations and improve the performance of earlier systems. The major feature of Spark that makes it unique is its ability to perform in-memory computations. It allows the data to be cached in memory, thus eliminating the Hadoop's disk overhead limitation for iterative tasks. Spark is a general engine for large-scale data processing that supports Java, Scala and Python and for certain tasks it is tested to be up to 100× faster than Hadoop MapReduce when the data can fit in the memory and up to 10× faster when data resides on the disk.[11]

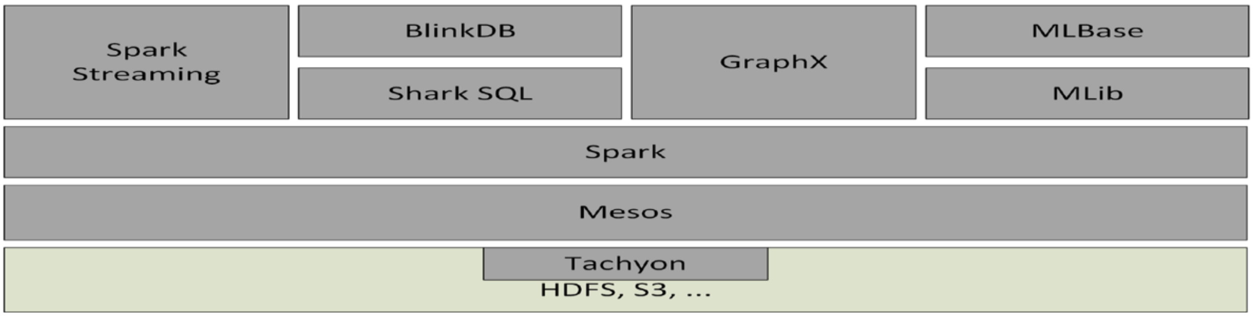

Fig-5 Spark Framework

It can run on Hadoop Yarn manager and can read data from HDFS. This makes it extremely versatile to run on different systems The Spark developers have also proposed an entire data processing stack called Berkeley Data Analytics Stack (BDAS) . At the lowest level of this stack, there is a component called Tachyon which is based on HDFS. It is a fault tolerant distributed file system which enables file sharing at memory-speed (data I/O speed comparable to system memory) across a cluster. It works with cluster frameworks such as Spark and MapReduce.

There are three ways of Spark deployment as explained below.

1. **Standalone:** Spark Standalone deployment means Spark occupies the place on top of HDFS(Hadoop Distributed File System) and space is allocated for HDFS, explicitly. Here, Spark and MapReduce will run side by side to cover all spark jobs on cluster.
2. **Hadoop Yarn:** Hadoop Yarn deployment means, simply, spark runs on Yarn without any pre-installation or root access required. It helps to integrate Spark into Hadoop ecosystem or Hadoop stack. It allows other components to run on top of stack.
3. **Spark in MapReduce (SIMR):** Spark in MapReduce is used to launch spark job in addition to standalone deployment. With SIMR, user can start Spark and uses its shell without any administrative access.

Spark offers an abstraction called Resilient distributed Datasets (RDDs) to support these applications efficiently. RDDs can be stored in memory between queries without requiring replication. Instead, they rebuild lost data on failure using lineage:[14] each RDD remembers how it was built from other datasets (by transformations like map, join or group By) to rebuild itself. RDDs allow Spark to outperform existing models by up to 100x in multi-pass analytics. RDDs can support a wide variety of iterative algorithms, as well as interactive data mining and a highly efficient SQL engine Shark [15].

## V. COMPARISON OF MAPREDUCE AND SPARK

This comparative study of Big data analytics Tools based on their parameters is listed below. The main motive of this comparison is to make the best selection of tool with respect to their areas.

| MAPREDUCE | SPARK |
|---|---|
| MapReduce is inefficient for multi-pass applications that require low-latency data sharing across multiple parallel operations. | Spark allows us to perform stream processing with large input data and deal with only a chunk of data on the fly. This can also be used for online machine learning, and is highly appropriate for use cases with a requirement for real time analysis |
| Slower as intermediate data/result is stored in hard disk | Up to 100 times faster to Hadoop, especially in iterative operations, as intermediate data/result is persist in memory.<br>persisted in memory |
| Mainly a batch processing engine where users can depend on other compatible platforms for performing stream processing, machine learning or database querying. | Spark being a batch processing engine also includes spark streaming for streaming data processing, MLLib for machine learning, GraphX for graph processing and spark SQL for querying thus providing an all-in-one<br>solution. |
| Lesser memory requirement | Memory requirement is higher. Degradation in performance if data not fit in the memory |
| Each Map task outputs the data in Key and Value pair. The output is stored in a CIRCULAR BUFFER instead of writing to disk. | The output of map side is written to OS BUFFER Cache. The operating system will decide if the data can stay in OS buffer cache or should it be spilled to Disk. |

Table-1 Mapreduce vs. Spark

## VI. CONCLUSION

This paper has given the brief introduction of Big data and analytics along with its features- Hadoop Mapreduce and apache Spark. This paper contains the chart comparison between these two tools. Each tool has its own advantages and disadvantages. By employing this comparative study, this concluded that spark has  better than Mapreduce . This comparative study will make things easier to the learner in the selection of Big data analytics tools according to their areas. In future, we will find out the solution how to improve the accuracy by applying machine learning algorithm on it.

## REFRENCES

[1] M.H.Padgavankar ,Dr.S.R.Gupta, Big Data Storage and Challenges, M.H.Padgavankar, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (2) , 2014, 2218-2223.

[2]. Sabia, Sheetal Kalra, Applications of big Data: Current Status and Future Scope, International Journal on Advanced Computer Theory and Engineering (IJACTE), , Volume -3, Issue -5, 2014, ISSN 2319-2526.

[3] Apache Hadoop. What Is Apache Hadoop?, 2014. http://hadoop.apache.org/, accessed April 2014.

[4] H S. Bhosale1, Prof. D. P. Gadekar2, A Review Paper on Big Data and Hadoop, International Journal of Scientific and Research Publications, 4(10),2014.

[5] C.Jin, R.Liu, Z.Chen, Alok  Choudhary, A Scalable Hierarchical Clustering Algorithm Using Spark, IEEE,

[6] Christos Doulkeridis, Kjetil ,  A Survey of Large-Scale Analytical Query Processing in MapReduce, The VLDB Journal manuscript No.5.

[7]. Lekha R.Nair, DR. Sujala,D.Shetty, streaming Twitter Data Analysis Using Spark For Effective  Job Search, Journal of Theoretical and Applied Information Technology ,. Vol.80. No. 2  2005 – 2015.

[8] Satish Gopalani,Rohan Arora, Comparing Apache Spark and Map Reduce with Performance Analysis using K-Means, International Journal of Computer Applications Volume 113 – No. 1, March 2015. (0975 – 8887)

[9] D. Rajasekar, C. Dhanamani, S. K. Sandhya,  A Survey on Big Data Concepts and Tools

[10] Suresh Lakavath, Ramlal Naik L, A Big Data Hadoop Architecture for Online Analysis, International Journal of Computer Science and Information Technology & Security (IJCSITS), Vol. 4, No.6, December 2014, ISSN: 2249-9555.

[11] M. Dhavapriya, N. Yasodha,  Big Data Analytics: Challenges and Solutions Using Hadoop, Map Reduce and Big Table, International Journal of Computer Science Trends and Technology (IJCST) – Volume 4 Issue 1, Jan - Feb 2016

[12] H.HU1, Y. WEN 2 , TAT-SENG CHUA1, AND XUELONG LI 3,  Toward Scalable Systems for Big Data Analytics, A Technology Tutorial, IEEE, 2 ,655-687, 2014.

[13] Ambika P R, Dr. K.N. Narasimha Murthy, Sowmya Naik PT, Aparna J S, Big Data: Towards Next Generation Analytics, International Journal of Innovative Research in Computer and Communication Engineering. Vol.3, Special Issue 5, May 2015.

[14]. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. Franklin, S. Shenker, and I. Stoica. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. Technical Report UCB/EECS-2011-82, EECS Department, University of California, Berkeley, 2011

[15] Reynold Xin, Joshua Rosen, Matei, Zaharia, Michael J. Franklin, Scott Shenker, Ion Stoica. Shark: SQL and Rich Analytics at Scale. SIGMOD 2013. June 2013.