International Journal of Current Engineering and Technology
©2017 INPRESSCO®, All Rights Reserved

E-ISSN 2277 – 4106, P-ISSN 2347 – 5161
Available at http://inpressco.com/category/ijcet

General Article

Study of Big Data Architecture and Tools

Mamta Mittal[*]

Department of Computer Science and Engineering, G B Pant Govt. Engineering College, New Delhi, India

Accepted 12 Jan. 2017, Available online 22 Feb 2017, **Vol.7, No.1 (Feb 2017)**

Abstract

Information technology has revolutionized our lifestyle, as a result abundant structured and non-structured data is now available. The rate of growth is of the order of zeta bytes and even further growing every year. These data sets are not only huge in size but also complex and diverse in nature. Existing data processing systems failed to analyse it. So, a new term Big Data came into existence which has provided new methodologies to manage and analyse these datasets. In this paper Big Data architecture and various management tools have been discussed in detail.

Keywords: *Big Data, Hadoop, Map Reduce, Spark*

1. Introduction

With the growth of data from social media, and other sources such as mobile devices and sensors, the traditional database management systems cannot handle such large and complex data. Therefore, Big Data algorithms can be used to store and manage the data and analyze it to get information. In the last two years, 90% of data in the world has been generated. Sources of Big Data are data from text messages, sensors, financial transactions, data from social media sites and mobile phones. As the amount of data increases exponentially the current database management techniques cannot deal with such large amount of data. Comprehensive coding skills and domain knowledge are required to deal with Big Data (Sagiroglu *et al* 2013).

In the academia and the IT industry, much attention has been given to Big Data. Rate at which data is generated is very high in case of Big Data. Currently, over 5 billion individuals use mobile phones and over 2 billion people use the Internet. The first characteristic that comes to our mind when we think of Big Data is size. In 2001, META Group analyst Doug Laney defined the 3V's of Big Data as Volume, Variety, and Velocity (Laney 2001). Gartner, Inc. defines Big Data: "Big Data is high-volume, high-velocity and high-variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making".

The "3Vs" model for describing big data are: ***Volume*** states to the huge volumes of data generated every second. ***Velocity*** states to the rate at which the data is produced. ***Variety*** refers to the different types of data means structured, unstructured or semi-structured data (Gandomi *et al* 2015). Nowadays six more V's have also contributed to big data: ***Veracity*** refers to trustworthiness of data. ***Value*** refers to the profit gained by organizations who invest in Big Data technologies. ***Variability*** refers to the inconsistencies in Big Data due to multiple sources of data. ***Validity*** is how accurate and correct Big Data is. ***Visualization*** refers to difficulty to visualize Big Data. ***Volatility*** is the time period after which data is not useful any longer (Gupta *et al* 2016).These 9Vs are described in Figure 1.

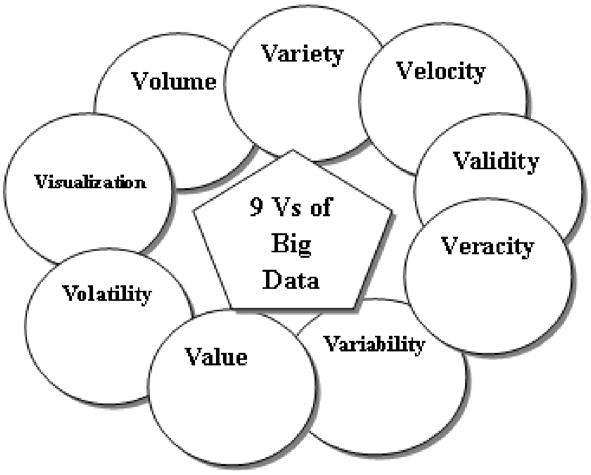

Figure1 9Vs of Big Data

[*] Corresponding author: **Mamta Mittal**

Big Data analytics finds its application in business, health, banking and agriculture etc. Big Data is used almost everywhere. In the present-day scenario Big Data is used ubiquitously (Chen *et al* 2014). The major applications of Big Data are banking, agriculture, marketing, telecom, health care food industry and fraud detection (Mukherjee *et al* 2016). The remaining paper has been organized as follows. Architecture of Big Data has been presented in Section 2. In Section 3 Big Data management tools Hadoop and Spark have been described. A brief conclusion has been given in last section.

2. Big Data Architecture

Big Data architecture defines how Big Data will be stored, managed and analyzed. It also defines the processing of Big Data components like database, storage used, software and hardware etc. The architecture is first created by the designers before physically implementing it. An understanding of business and organization needs for Big Data is required for creating Big Data architecture. Figure 2 illustrates the general architecture of Big Data.

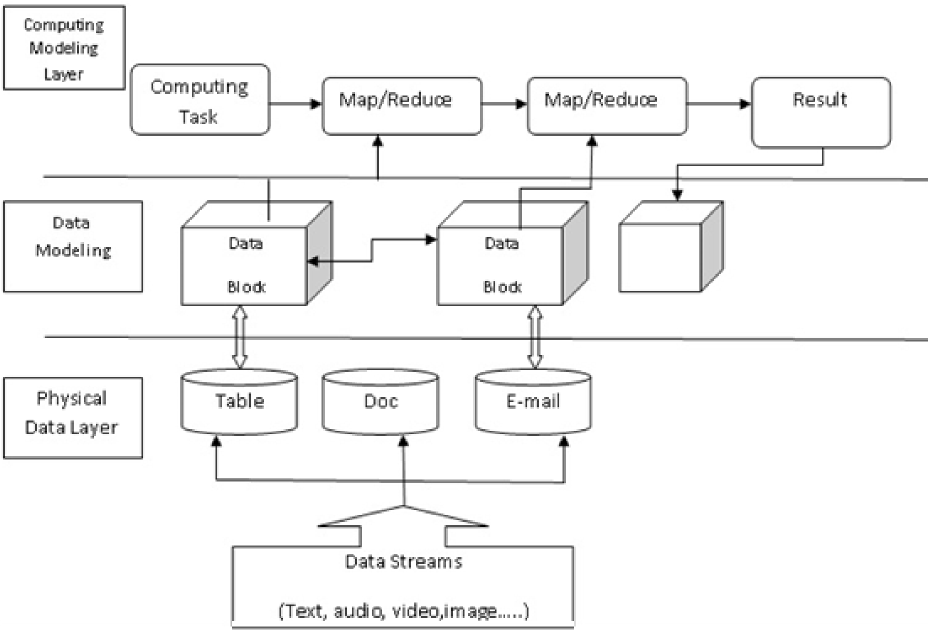

Figure 2 Architecture of Big Data

The architecture of Big Data consists of three model layers:

- The physical data layer takes data in the form of text, video, audio, images etc.
- The data modeling layer is the layer that is used to handle physical data.
- The computing modeling layer is used to convert data into information so that decision making can be done.

3. Big Data Management Tools

Due to the evolution of computing technology, large amount of data can be managed without using high cost and supercomputers. Big Data cannot be handled by a single computer and its structure is different from traditional data. Therefore, some tools and technologies are required to store, manage, and analyze Big Data in real time. For Big Data, some of the most commonly used tools are Hadoop and Spark which are described in the following section in details:

3.1 Hadoop

Hadoop is an Apache project which started in the year 2006. Developed by Doug Cutteing, Hadoop is an open-source framework that stores and processes Big Data by distributing data across clusters of nodes (Khan *et al* 2014). Hadoop is based on Google's Map Reduce programming environment. Hadoop is consisting of Pig, Hive, HBase, HCatalog, Oozie, Zookeeper, and Kafka. The two most important components of Big Data are Map Reduce and Hadoop Distributed File System (HDFS). In Hadoop, data is stored in HDFS and Map Reduce is used to process data in parallel on different nodes. It is capable of performing analysis for large amount of data (Dittrich *et al* 2013). The architecture of Hadoop is shown in Figure 3.

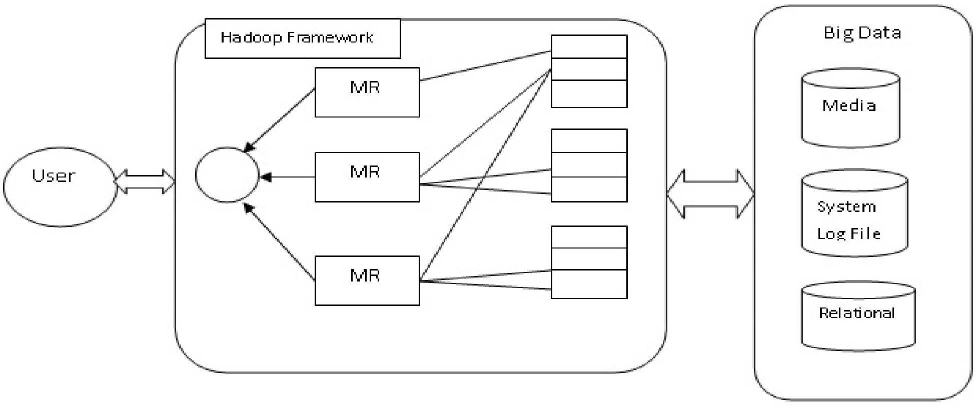

Figure3 Architecture of Hadoop

Hadoop consists of the following modules:

- *Hadoop Common*: They consist of Java libraries required by other Hadoop modules. They provide the distributed file system and the necessary Java files and scripts that are needed to start Hadoop.
- *Hadoop YARN*: Cluster management and job scheduling is done using this module.
- *Hadoop Distributed File System (HDFS)*: It is similar to Google File System where the application data is stored.
- *Hadoop Map Reduce*: It is used for processing of large data sets in parallel on different nodes.

HDFS is based on the Google File System (GFS). It is a file system that is designed to run on single node and multi node clusters. In a multi node cluster, HDFS has a master/slave architecture. Master node consists of a single **Name Node** that has a job tracker which stores the file metadata that is information about data and one or more slave and **Data Nodes** that has task tracker which stores the actual data. For a single node cluster, Name Node and Data Node are present on the same machine. HDFS splits a file into various HDFS blocks. Copy of these blocks is sent to at least three Data Nodes. This mapping to different Data Nodes is determined by the Name Node. Block creation, deletion and replication is done by the Data Node. It performs the read and write operation in the file system. HDFS provides a shell and a set of commands to interact with the file system. Hadoop Map Reduce is a framework which processes huge amount of data by distributing it on large clusters. Map Reduce can be divided into two stages:

- *Map Task*: It converts the input data into different datasets. The elements of the dataset are broken down into key value pairs. The result from the map task is stored in the local file system so that reducer can access it.
- *Reduce Task*: It converts the output from a mapper into a smaller set of tuples. The Job Tracker monitors the reducer tasks which are executed on the worker nodes.

In Map Reduce Framework Job Tracker performs different tasks such as job scheduling, monitoring jobs and rescheduling the failed jobs. Based on the master' s instructions, the tasks are performed by the slave Task Tracker. It also provides the status information of the task to the master periodically.

3.2 Spark

Spark was developed at UC Berkeley's AMPLab in the year 2009 (Jonathan 2015). It is an open source Hadoop engine that is intended for fast execution of large data sets. Resilient Distributed Dataset (RDD) is known as the programming interface of Spark. It is a read-only set of data structure that is distributed over a number of nodes. Spark overcomes a limitation in Map Reduce, where the dataflow on distributed programs is linear: Map Reduce programs convert the data into a number of sets and then a map function is performed. The results of the map are reduced, and the results are stored on the disk. In Spark's RDDs, the distributed shared memory offered is restricted. In RDD, iterative algorithms have been implemented, which visit their dataset again and again, and perform interactive data analysis. Therefore, the latency is reduced in this as compared to Hadoop. The Spark architecture consists of a cluster manager and a distributed storage system.

- Spark consists of the native Spark cluster, Hadoop YARN and or Apache for the cluster management.
- Spark can be used with HDFS, MapR File System , Cassandra, Amazon S3 for the purpose of distributed storage.

Conclusion

In this paper basic architecture and most popular tools Hadoop and Spark has been discussed in details. From this study it is concluded that Spark tool is helpful in real time data as it solves fast interactive queries in seconds and can handle much more and variety of operations in comparison with Hadoop. It is compatible with any Hadoop storage system like HDFS and HBase.

References

Sagiroglu, S., & Sinanc, D. (2013, May). Big data: A review. In *Collaboration Technologies and Systems (CTS), 2013 International Conference on* (pp. 42-47). IEEE.

Dittrich, J., Richter, S., & Schuh, S. (2013). Efficient or Hadoop: why not both?. *Datenbank-Spektrum*, *13*(1), 17-22.

Chen, M., Mao, S., & Liu, Y. (2014). Big data: a survey. *Mobile Networks and Applications*, *19*(2), 171-209.

Khan, N., Yaqoob, I., Hashem, I. A. T., Inayat, Z., Mahmoud Ali, W. K., Alam, M., & Gani, A. (2014). Big data: survey, technologies, opportunities, and challenges. *The Scientific World Journal*, *2014*.

Gandomi, A., & Haider, M. (2015). Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*, *35*(2), 137-144.

Jonathan Buckley, (2015). *Apache Spark vs. Hadoop Which Big Data Framework is the Best Fit?* Retrieved December 17, 2015 from https://www.qubole.com/blog/big-data/spark-vs-hadoop

Mukherjee, S., & Shaw, R. (2016). Big Data–Concepts, Applications, Challenges and Future Scope*. International Journal of Advanced Research in Computer and Communication Engineering Vol.* 5, Issue 2.

Laney, D. (2001). 3D data management: Controlling data volume, velocity and variety. *META Group Research Note*, *6*, 70.

Gupta, P., Sharma, A., & Jindal, R. (2016). Scalable machine-learning algorithms for big data analytics: a comprehensive review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *6*(6), 194-214.