A Survey on Big Data, Hadoop and it's Ecosystem

Jyotsna Y. Mali
Asst. Prof. in CSE
T.K.I.E.T., Kolhapur, India
jyotsmali@gmail.com

Abhaysinh V. Surve
Asst. Prof. in CSE
T.K.I.E.T., Kolhapur, India
avsurve@tkietwarana.org

Vaishali R. Khot
Asst. Prof. in CSE
T.K.I.E.T., Kolhapur, India
vaishalikhot25@gmail.com

Anita A. Bhosale
Asst. Prof. in CSE
T.K.I.E.T., Kolhapur, India
bhosale.anita11@gmail.com

***Abstract*:** Now days, The 21st century is emphasized by a rapid and enormous change in the field of information technology. It is a non-separable part of our daily life and of multiple other industries like education, genetics, entertainment, science & technology, business etc. In this information age, a vast amount of data generation takes place. This vast amount of data is referred as Big Data. There is a number of challenges present in the Big Data such as capturing data, data analysis, searching of data, sharing of data, filtering of data etc. Today Big Data is applied in various fields like shopping websites such as Amazon, Flipkart, Social networking sites such as Twitter, Facebook, and so on. It is reviewed from some literature that, the Big data tends to use different analysis methods, like predictive analysis, user analysis etc. This paper represents the fact that, Big Data required an open source technology for operating and storing huge amount of data. This paper greatly emphasizes on Apache Hadoop, which has become dominant due to its applicability for processing of big data.Hadoop supports thousands of terabytes of data. Hadoop framework facilitates the analysis of big data and its processing methodologies as well as the structure of an ecosystem.

Keywords: *Big Data, Ecosystem, Hadoop, HDFS, Map Reduce, MasterNode, NameNode*

*****

I. Introduction

Now days, competitions are rapidly increasing so organizations must possess a number of skills to create their position and remain alive in the market. The data generated from mobile devices, credit cards, social networking platforms may remain unused on unknown servers for many years. Big data may be useful to assess and analyze this data to generate necessary information. Some real-time examples of Big Data are, Shopping websites such as Amazon, Flipkart, Social networking sites such as Twitter, Facebook, and so on. Big Data is structured, semi-structured or unstructured in nature. It is not possible for traditional data management, warehousing, and analysis systems to analyze the huge amount of data [2]. Due to this problematic situation, Big Data is stored in distributed architecture file systems.

A. Elements of Big Data:

Gartner states that, data increases at the rate of 59% per year. This is represented in terms of following 4 V's: Volume, Velocity, Variety and Veracity.

- ***Volume:*** Volume is the quantity of data generated by organizations or individuals. Today the range of data in organizations is coming in exa-bytes. For example, The Internet alone generates a huge amount of data, it is having approximately 14.3 trillion web pages, 672 exa-bytes of accessible data, over 9,00,000 servers are owned by Google which is the largest in the world.
- ***Velocity:*** It specifies the rate at which data is generated, captured and shared. The sources of high velocity of data includes IT devices like routers, switches, firewalls, Social medias containing Facebook, Twitter etc.
- ***Variety:*** data comes from various sources such as internal, external. Data comes in no of formats such as text, video, image, audio etc. GPS, Social networking sites can generate different formats of data.
- **Veracity:** the obtained data is correct or consisting but the Veracity. this uncertainty is NOTHING but the Veracity

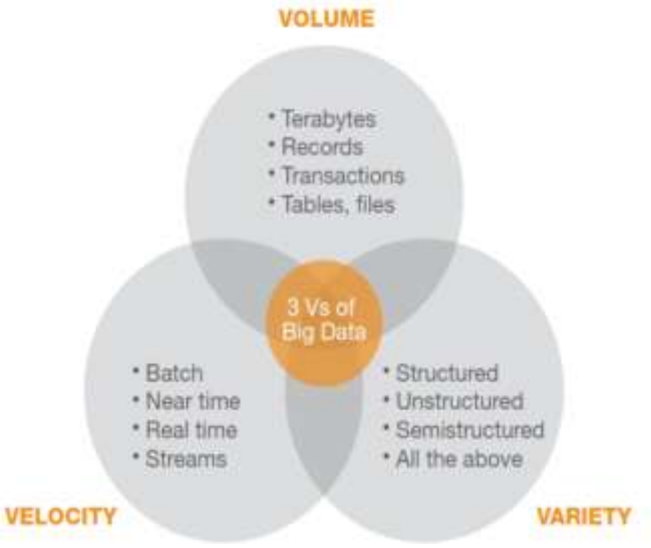

Figure 1: Elements of Big Data

II. Hadoop

An open-source software distributed storage and distributed processing of extremely large data sets on computer clusters is Hadoop. Massive storage of data, enormous processing power and ability to handle virtually limitless concurrent tasks is facilitated by hadoop .

Hadoop runs applications on systems with thousands of commodity hardware nodes, and handles thousands of terabytes of data. Its distributed file system facilitates fast data transfer rates among nodes and allows the system to continue operating in case of a node failure. Large amount of data can be handled by Hadoop like images ,audio,video, sensor communications, folder, files etc. [3]. Hadoop was created by computer scientists Doug Cutting and Mike Cafarella in 2006 to support distribution for the Nutch search engine. It was inspired by Google's MapReduce, a software framework in which an application is broken down into numerous small parts. Any of these parts, which are also called fragments or blocks, can be run on any node in the cluster. After years of development within the open source community, Hadoop 1.0 became publically available in November 2012 as part of the Apache project sponsored by the Apache Software Foundation.

From its initial release, Hadoop has been developed and updated. The second iteration of Hadoop (Hadoop 2) enhanced resource management and scheduling. It features a high-availability file-system option and support for Microsoft Windows and other components to expand the framework's versatility for data processing and analytics. The core of Apache Hadoop consists of a storage part, known as Hadoop Distributed File System (HDFS), and a processing part called MapReduce.

As a software framework, Hadoop is composed of numerous functional modules. At a minimum, Hadoop uses Hadoop Common as a kernel to provide the framework's essential libraries. Other components include Hadoop Distributed File System (HDFS), which is capable of storing data across thousands of commodity servers to achieve high bandwidth between nodes; Hadoop Yet Another Resource Negotiator (YARN), which provides resource management and scheduling for user applications; and Hadoop MapReduce, which provides the programming model used to tackle large distributed data processing -- mapping data and reducing it to a result [1].

A. Hadoop Distributed File System (HDFS):

HDFS is designed to manage the challenges of Big Data. HDFS is represented as master slave architecture having a single NameNode and more than one DataNodes. The NameNode manages the file system and hold all of its metadata in RAM. The namenode knows the data nodes on which all the blocks for a given file are located. DataNodes act as the worker nodes of the file system [4]. A file is spitted into one or more blocks (default 64MB or 128MB) and that blocks are stored in DataNodes. Secondary Name node communicates with the NameNode to take checkpoints of the HDFS metadata at intervals defined by the cluster configuration but it is not a backup of NameNode. When it is required, the primary NameNode reads the check pointed image. It is usually run on a different server than the primary NameNode. It provides monitoring the state of the cluster HDFS and each cluster has one Secondary Name Node.

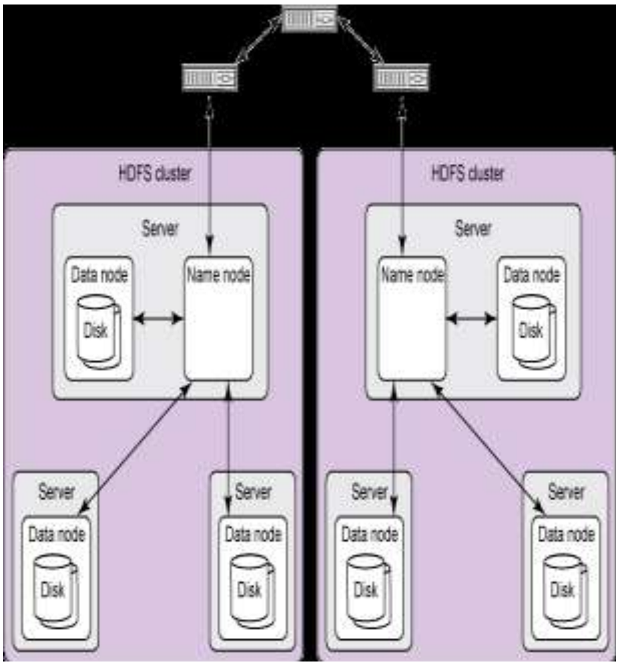

Figure 2: HDFS Architecture and Daemons

1) Features of HDFS:

- It is suitable for the distributed storage and processing.
- Hadoop supports a command interface to interact with HDFS.
- The advanced built servers of namenode and datanode help users to easily check the status of cluster.
- file system data is having streaming access.
- file permissions and authentication are provided by HDFS.

2) Concept of Block in HDFS Achitecture:

A disk has a certain block size, which is the basic measure of information that it can read or compose. File systems expand by managing information in pieces, which are an indispensable part of the disk block size [1].HDFS blocks are very large in contrast to disk blocks . Consequently, the time to transfer a huge record made of multiple blocks operates at the disk exchange rate.

3) NameNodes and DataNodes:

An HDFS cluster has two node types working in a slave master design: a NameNode (The Master) and various DataNodes (Slaves). The NameNode deals with the file system. It Stores the metadata for all the documents and indexes in the file system. This metadata is stored on the local disk as two files: the file system and edit log [5]. The NameNode is aware of the DataNodes on which all the pieces of a given document are found; however, it doesn't store block locations necessarily, since this data is recreated from DataNodes.

The file system is accessed by client on behalf of the user by communicating with the DataNodes and NameNodes. DataNodes are the workhorses of a file system. They store and recover blocks when they are asked to(by clients or the NameNode), and they report back to the Namenode occasionally with a list of blocks that they store externally. DataNodes connect with the NameNode by sending messages.

B. MapReduce:

The MapReduce software framework which was originally introduced by Google in 2004 is a programming model, which now adopted by Apache Hadoop, consists of splitting the large chunks of data and 'Map' & 'Reduce' phases. Map reduce is a processing large datasets in parallel using lots of computer running in a cluster [1]. We can extend the mapper class with our own instruction for handling various input in specific manner. During map master computer instructs worker computers to process local input data and Hadoop performs shuffle process. Thus master computer collects the results from all reducers and compilers to answer overall query.

The major advantage of MapReduce is that it is easy to scale data processing over multiple computing nodes. Under the MapReduce model, the data processing primitives are called mappers and reducers. Decomposing a data processing application into mappers and reducers is sometimes nontrivial. But, once we write an application in the MapReduce form, scaling the application to run over hundreds, thousands, or even tens of thousands of machines in a cluster is merely a configuration change. This simple scalability is what has attracted many programmers to use the MapReduce model.

Generally MapReduce paradigm is based on sending the computer to where the data resides MapReduce program executes in three stages, namely map stage, shuffle stage, and reduce stage.

- ***Map Stage****:* In this stage the strategy starts by identifying the amount of memory [6].The map or mapper's job is to process the input data. HDFS stores the input data in the form of files and directory The input file is passed to the mapper function line by line. The mapper processes the data and creates several small chunks of data.
- ***Reduce Stage***: This stage is the combination of the Shuffle stage and the Reduce stage. The Reducer's job is to process the data that comes from the mapper. After processing, it produces a new set of output, which will be stored in the HDFS.
- During a MapReduce job, Hadoop sends the Map and Reduce tasks to the appropriate servers in the cluster [1].
- The framework manages all the details of data-passing such as issuing tasks, verifying task completion, and copying data around the cluster between the nodes.
- Most of the computing takes place on nodes with data on local disks that reduces the network traffic.
- After completion of the given tasks, the cluster collects and reduces the data to form an appropriate result, and sends it back to the Hadoop server.

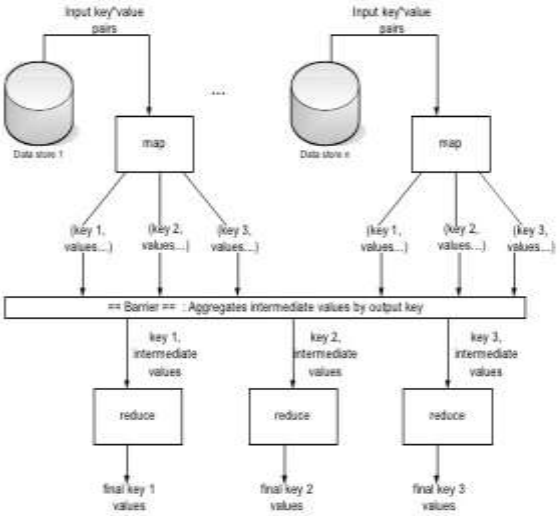

Figure 3: Traditional MapReduce workflow

III. HADOOP ECOSYSTEM:

Hadoop ecosystem is a framework of various types of complex and evolving tools and components. In simple words, Hadoop ecosystem is a collection of tools and technologies that can be effectively implemented and deployed to provide Big Data solutions in a cost-effective manner. **MapReduce** and **Hadoop Distributed File System (HDFS)** are two components of the Hadoop ecosystem that provide a great starting point to manage

Big Data [7]. Along with these two, the Hadoop ecosystem provides a collection of various elements to support the complete development and deployment of Big Data solutions. Figure 4 depicts the elements of the Hadoop ecosystem.

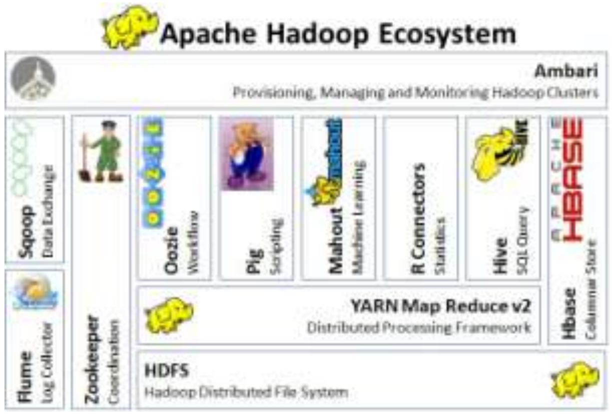

Figure 4 Hadoop Ecosystem [Source:http://blog.agro-know.com/?p=3810]

All these elements enable users to process large datasets in real time and provide tools to support various types of Hadoop projects, Schedule jobs, and manage cluster resources. Figure 5 depicts how the various elements of Hadoop involve at various stages of processing data:

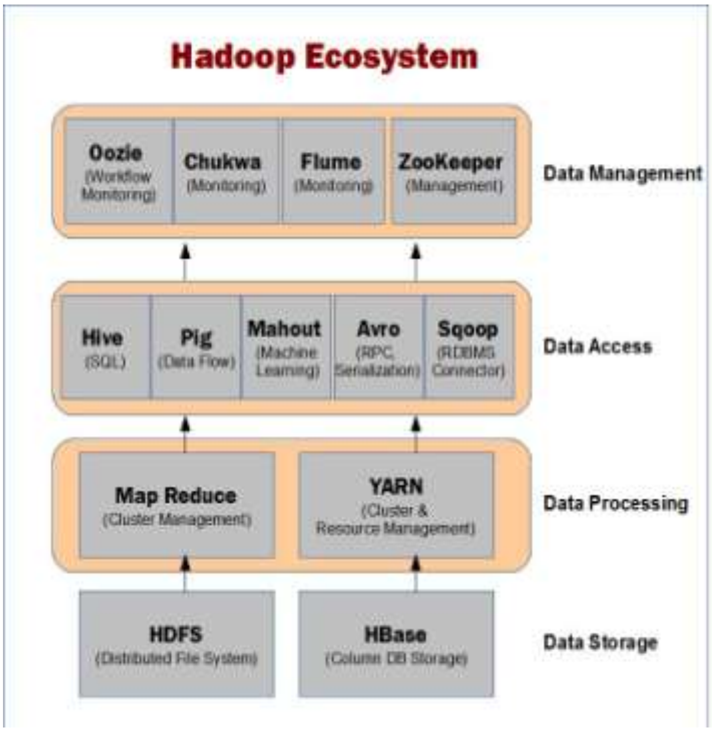
Hadoop Ecosystem
Oozie (Workflow Monitoring)
Chukwa (Monitoring)
Flume (Monitoring)
ZooKeeper (Management)
Data Management
Hive (SQL)
Pig (Data Flow)
Mahout (Machine Learning)
Avro (RPC, Serialization)
Sqoop (RDBMS Connector)
Data Access
Map Reduce (Cluster Management)
YARN (Cluster & Resource Management)
Data Processing
HDFS (Distributed File System)
HBase (Column DB Storage)
Data Storage

Figure 5 Hadoop Ecosystem Elements at Various Stages of Data Processing

A. Hive:

Hive is a data warehousing package built on top of Hadoop and SQL like access for data in HDFS.It provides an interface, similar to SQL , which enables you to create databases and tables for storing data [7]. In this way, you can achieve MapReduce concept without explicitly writing the source code for it.

Hive provides a Structured Query Language (SQL) interface, HiveQL (Hive Query Language). This interface translates the given query into a MapReduce code. HiveQL enables users to perform tasks using the MapReduce concept but without explicitly writing the code in terms of the map and reduce functions. The data stored in HDFS can be accessed through HiveQL, which contains the features of SQL but runs on the MapReduce framework.

B. Pig and Pig Latin:

The MapReduce model is not always convenient and efficient, but it can solve many real-world problems. However, when there is a large amount of data that needs to be processed using Hadoop, the processing involves more overhead and becomes complex. A better solution to such kind of problems is Pig, which is a Hadoop extension. Pig handles gigabytes or terabytes of data.

Data operations are performed by Pig. The Pig platform is specially designed for handling many kinds of data be it structured, semi-structured or unstructured. Pig was developed in 2006 at Yahoo. It was a part of research project, the aim of which was to provide an easy option to simplify the process of using Hadoop and concentrate on examining large datasets instead of wasting time on MapReduce. Pig became an Apache Project in 2007. In 2009, Pig started being used by other companies and emerged as a top-level Apache project in 2010.

Pig consists of a scripting language, known as Pig Latin, and a Pig Latin compiler. The scripting language is used to write the code for analyzing the data, and the compiler converts the code into the equivalent MapReduce code.

C. YARN:

Most Big Data architectures are based on Hadoop ecosystem, which has evolved rapidly over the years. The first version of Hadoop ecosystem (Hadoop 1) included MapReduce data processing model, which provides limited scalability and performance because of the batch-oriented nature of processing the data. The MapReduce model improved over time to become more interactive and specialized in Hadoop 2. Among the various improvements of the Hadoop 2.0, Yet Another Resource Negotiator (YARN) is one of the most important improvements. YARN is a key element of the Hadoop data processing architecture that provides different data handling mechanisms, including interactive SQL and batch processing. It improves the performance of data processing in Hadoop by separating the resource management and scheduling capabilities of MapReduce from its data processing component. YARN has more flexible data processing engine and supports additional processing models such as Bulk Synchronous and Parallel (BSP) model. Thus, YARN can be considered as an operating system of Hadoop ecosystem, because it is responsible for managing, monitoring, and maintaining a multi-tenant environment.

D. Flume:

A large amount of data is required for analytical processing, and data is loaded from different sources into Hadoop clusters. The process of loading huge data into Hadoop clusters from different sources faces problems like maintaining and ensuring data consistency and the best way of utilizing the resources.

Flume is a framework developed by Apache developers keeping in mind that it should be capable of solving the recursively generating data, i.e., logs, crash reports, etc. Flume is used to collect data from different sources, such as logs, directories, etc. It scales data horizontally, and multiple flume agents can be put in action to collect large volumes of data from different sources and aggregate them. Thereafter, data logs are moved to a centralized data store, i.e. HDFS. Apache Flume is distributed, reliable and has a flexible architecture based on streaming data flows.

E. Sqoop:

Sqoop is a tool designed to exchange information between Hadoop and relational database. As the compositional focus of Apache Hadoop, YARN coordinates information ingested from Apache Sqoop and different administrations that convey information into the Enterprise Hadoop group. Apache Sqoop effectively exchanges mass information between Apache Hadoop and organized data stores (e.g. relational databases). It performs with relational databases such as Teradata, Netezza, Oracle, MySQL, Postgres and HSQLDB.

F. ZooKeeper:

Hadoop works by the divide-and-conquer approach. Once a problem is divided, it is approached and processed by using distributed and parallel processing techniques across Hadoop clusters. In case of Big Data problems, traditional interactive tools do not provide enough insight or timelines required to take business decisions. In that case, Big Data problems are approached with distributed applications. ZooKeeper helps in coordinating all the elements of the distributed applications.

G. Oozie:

A team of Yahoo developers realized the scope of Hadoop-based data processing systems and developed Oozie. Oozie is a new workflow management and scheduling system. It was first released in 2010 as the GitHub project. It supports the workflow / coordination model and is highly extensible and scalable.

IV. CONCLUSION

Hadoop is the platform for large amount of data processing. It solves the problem in greater extent related to Big Data. It is consisting of two main components HDFS and Map Reduce. Hadoop stores the data in distributed file system. Hadoop increases the speed of data processing and storage.

REFERENCES

[1] DT Editorial Services, “Big Data(covers Hadoop2, Map Reduce, Hive, Yarn, Pig, R and Data Visualization)” by Dreamtech Press

[2] Natalia Miloslavskaya ,Alexander Tolstoy, “Big Data, Fast Data and Data Lake Concepts” 7th Annual International Conference on Biologically Inspired Cognitive Architectures, Volume 88, 2016, Pages 300–305

[3] B. Saraladevi, N. Pazhaniraja, P. Victer Paul, M.S. Saleem Basha, P. Dhavachelvan, “Big Data and Hadoop-A Study in Security Perspective,” 2nd International Symposium on Big Data and Cloud Computing (ISBCC’15), Procedia Computer Science 50 (2015) 596 – 601

[4] “Hadoop, MapReduce and HDFS:A developer perspective,”(Procedia Computer Science, Volume 48, 2015,Pages 45-50)

[5] A Novel and efficient de-duplication system for HDFS(Procedia Computer Science ,Volume 92,2016, Pages (498-505)

[6] Tharso Ferreira, Antonio Espinosa, Juan Carlos Moure, Porfidio Hern´andez, “An Optimization for MapReduce Frameworks in Multi-core,” International Conference on Computational Science, ICCS 2013, Procedia Computer Science 18 (2013) 2587 – 2590

[7] Can Uzunkaya, Tolga Ensari, Yusuf Kavurucu, “Hadoop Ecosystem and Its Analysis on Tweets,” World Conference on Technology, Innovation and Entrepreneurship, Procedia - Social and Behavioral Sciences 195 (2015) 1890 – 1897

[8] Sachin Bende, Rajashree Shedge, “Dealing with Small Files Problem in Hadoop Distributed File System,” 7th International Conference on Communication, Computing and Virtualization 2016, Procedia Computer Science 79 (2016) 1001 – 1012

[9] Mouad Lemoudden, Meryem Amar, Bouabid El Ouahidi, "A Binary-Based MapReduce Analysis for Cloud Logs," Second International Workshop on Mobile Cloud Computing Systems, Management, and Security, Procedia Computer Science 83 (2016) 1213 – 1218

[10] Naveen Garg , Dr. Sanjay Singla, Dr. Surender Jangra, "Challenges and Techniques for Testing of Big Data," Procedia Computer Science 85 (2016) 940 – 948

[11] PekkaPääkkönen, DanielPakkala1, "Reference Architecture and Classification of Technologies, Products and Services for Big Data Systems,"