SECURE HADOOP ECOSYSTEM FOR BIG DATA

[1] Zakiullah, [2] Anjali Chaudhary
[1] M.Tech (SE), School of Engineering & Technology, Noida International
University, Plot 1, Yamuna Expy, Sector 17A, Uttar Pradesh 203201
[2] Assistant Professor, School of Engineering & Technology, Noida International University, Plot 1, Yamuna Expy, Sector 17A,
Uttar Pradesh 203201

Abstract: The management of enormous quantity of ever-growing digital data is an important topic today. This is especially important with regard to safe data storage, processing and organization. Various subsets of machines are generated in Apache Hadoop's ecosystem and data is then effectively coordinated between the users. In this paper we study the vital components of Hadoop ecosystem, Hadoop Distributed File System (HDFS) and MapReduce. HDFS is used as the main storage system for various Hadoop applications. HDFS offers high and robust data availability to various end user applications. MapReduce is an application environment for the analysis and transformation of big datasets into a specific outcome. This paper also focused on the examination and the various potentials and benefits of Hadoop with relation to HDFS 1.0, HDFS 2.0 and its analytical and security capabilities.

Keywords: Hadoop, Big data, Apache Hadoop's ecosystem

1. INTRODUCTION

Big data refers to large amount of data sets which may be unstructured, structured or semi-structured. Structured data means the data has a predefined data model or structure and are usually maintained using the traditional techniques. They include the data maintained in relational database, XML files etc. Unstructured data refers to the data that do not a have a well- defined model or structures which includes images, email etc.

Big data is everywhere - lots of data is being collected and warehoused. Data can be from science experiments, Web Data, Bank/credit transactions, Social Networks etc. These large data sets which are produced cannot be stored, analyzed, and processed using the traditional methods. The Big data "size" is now changing from terabytes to peta bytes of data.

One of the major challenges in big data storage and analysis is that the data can not be processed in parallel easily. With parallelism transfer speed improves at a greater rate than the seek speed. Parallelism is hard due to the following reasons: Synchronization, Limited bandwidth, Computer are complicate - Driver failure, Data Availability etc.

Another solution provided to overcome these problems was Distributed Computing. Distributed Computing deals with the study of distributed system in which different systems communicate with each other to achieve a specific goal.

1.1 Characteristics of Big data:

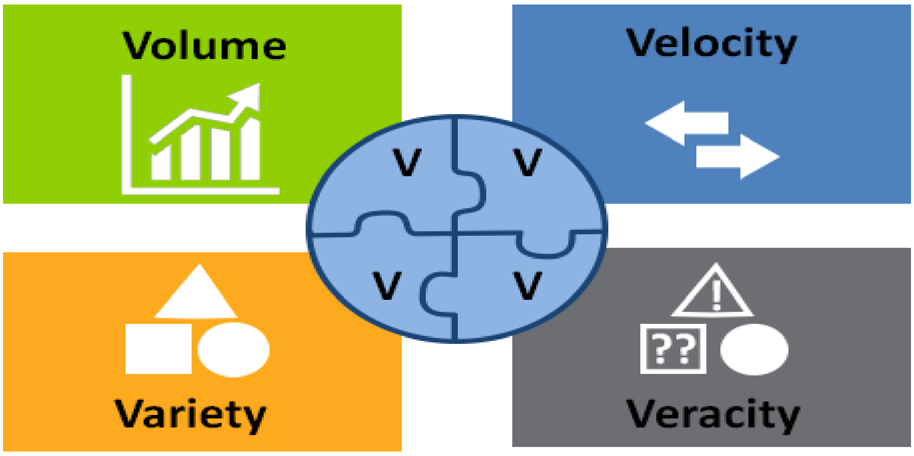

Figure 1: four V’s of big data

(i) Volume – The name Big Data itself is related to a size which is large. Size of data plays a very important role in determining value out of data. Also, whether a particular data can actually be considered as Big Data or not, is dependent upon the volume of data. Hence, **'Volume'** is a characteristic which needs to be considered while dealing with Big Data.

(ii) Velocity – The term 'velocity' refers to the speed of data processing. Big Data Velocity deals with the speed at which data flows in from sources such as business processes, application logs, networks, and social media sites, sensors, Mobile devices, etc. The flow of data is very large and continuous.

(iii) Variety – Variety refers to heterogeneous sources and the nature of data, structured and unstructured. In past days, spreadsheets and databases were only sources of data considered by a lot of applications. Nowadays data in the form of emails, photos, videos, monitoring devices, PDFs, etc are also being considered in the analysis applications.

(iv) Veracity – Last, but certainly not least that is veracity. Veracity is the quality of the data. Just how accurate is all this data? For example, think about all the Twitter posts with hash tags, abbreviations, etc., and the reliability and accuracy of all that content.

2. PAPER OBJECTIVES

In this paper we review the implementation of Hadoop ecosystem for big data. As Hadoop is not a single application but a collection of various components compiled under Google's MapReduce, therefore, it is necessary to understand various components of the Hadoop ecosystem and the challenges in Big data processing. The main objectives of the study were:

1. To analyse the successful implementation of Hadoop concept in the distributed system and data storage and distribution of Big data.

2. Understanding various framework components involved in the Hadoop platform and their role in processing of Big Data.

3. Analysing the security concerns under the Hadoop eco-system in context of handling Big data.

[1] "Huang, Nicol & Campbell (2014) focused on distributed environment security where YARN or Hadoop are used to process big data. They concentrated on the threat to security known as DoS, providing useful observations linked to the DoS attack effect." [3] "Lee and Lee utilized a scalable internet traffic measurement framework for Hadoop MapReduce. The unified engine identified as Apache Spark has been studied by Zahariaetal. (2016) for huge data processing. This platform provides the capabilities of big data processing to give organizations a great deal of value."

3. ORIGIN OF HADOOP

The works in open source Hadoop ecosystem along with licensed distribution have shown considerable guarantee to resolve many big data requirements in recent years (Apache Hadoop). The aim is to have the Hadoop Eco-system go far beyond the rendering paradigm of MapReduce and enable other software development paradigms to run on the same HDFS cluster to scale the above requirements (Demchenko & Membrey, 2013; Hahanov et al. 2015; Pääkkönen & Pakkala, 2015). This makes Hadoop ecosystem as big data platform a natural choice for becoming the heart of the enterprise big data platform. Figure 1 shows structure of the 2 versions of Hadoop (Hadoop 1.0 and Hadoop 2.0).

- Apache Hadoop is basically a framework that allows distributed processing of huge amount of datasets across multiple computer clusters using programming model.
- Hadoop is an open-source implementation of Google MapReduce and GFS.
- Developed by Doug Cutting.
- Hadoop fulfilled the need of common infrastructure by making it efficient, reliable ,easy to use, Open source etc.

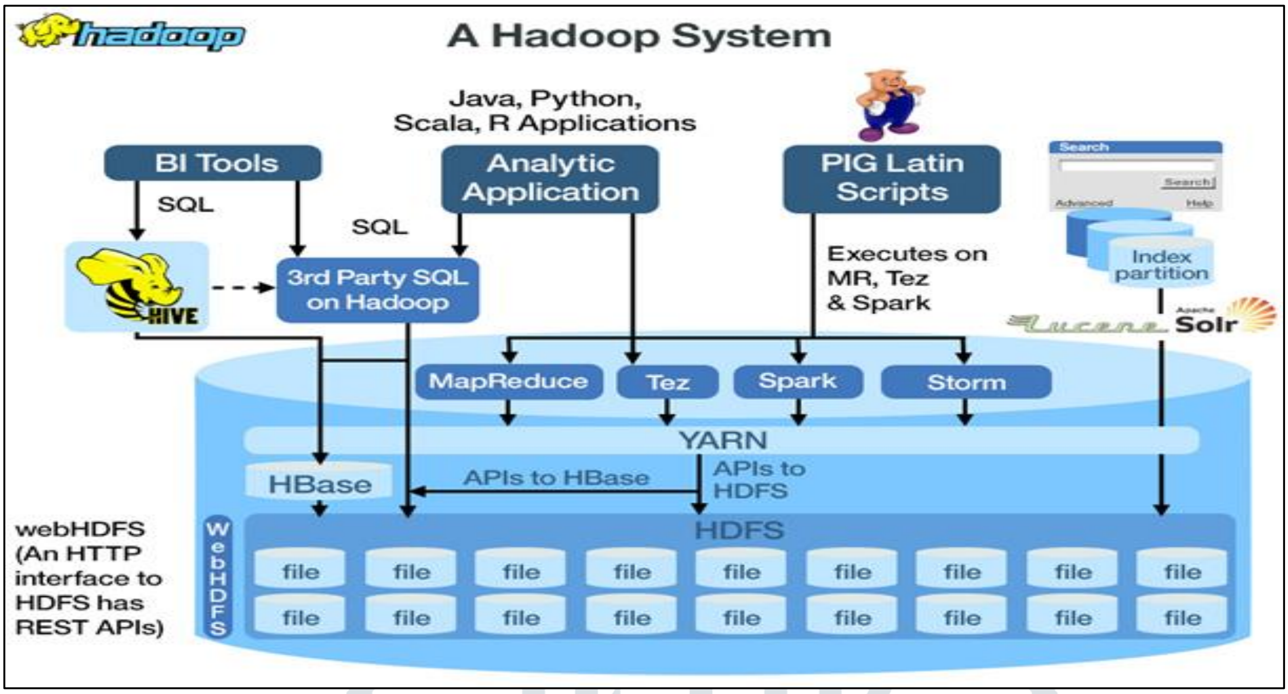

Figure 2: A Hadoop System

3.1 Limitations With Hadoop 1.X

1. In HADOOP 1.0,the MapReduce component supports only batch processing of large amount of data and does not support processing of real0 time data.

2. It cannot be used for data streaming.

3. Can support up to 4000 nodes per cluster.

4. The Jobtracker is the only single component present and has the following functions :

- Managing the resource for the nodes.

- Scheduling the jobs for the nodes to perform .

5. Only the Map/Reduce jobs can be run.

6. Only one Namenode is supported that is one namespace for each cluster.

7. Scalability is not provided in the horizontal direction.

8. In HDFS, slots concept is followed to allocate the resources. Therefore there are static Map and reduce slots. Once these are allocated to the Map/Reduce jobs, they cannot be later on used by the idle slots.

3.2 Limitations Overcame By Hadoop 2.X

1. Supports multiple Namespaces.

2. It uses different sizes of container rather than fixed slots in HADOOP 1.x

3. Supports 10,000 nodes / cluster.

4. It introduced the YARN component for the management of resources.
5. The MapReduce responsibilities were given to other components to reduce its overload.
6. Scalability is provided in the horizontal direction.

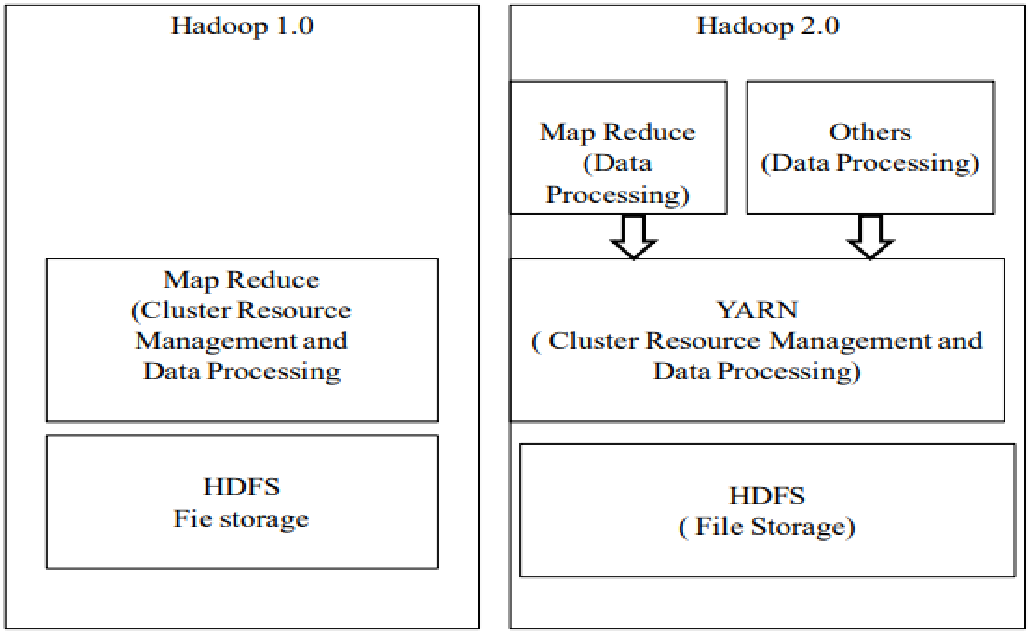
Hadoop 1.0
Map Reduce (Cluster Resource Management and Data Processing
HDFS Fie storage
Hadoop 2.0
Map Reduce (Data Processing)
Others (Data Processing)
YARN (Cluster Resource Management and Data Processing)
HDFS (File Storage)

Figure 3: Structure of Hadoop 1.x and Hadoop 2.x

The Hadoop Ecosystem comprises of **4** major components:

1. **Hadoop Common** -Basically consists of the libraries that are used by the other Hadoop modules . For example provides the libraries for the file system and OS level abstractions for the Hadoop to get started.
2. **Hadoop YARN** -YARN framework main purpose is to schedule jobs and management of cluster resources.
3. **Hadoop Distributed File System (HDFS)**-Provides high throughput access to application data.
4. **Hadoop MapReduce** – It is a YARN based system. Used for parallel processing of huge data sets.

4. HADOOP COMPONENTS AND ARCHITECTURE

Apache Hadoop software library can detects and handles failures at the application layer hence deliver high availability services in the top of all multiple clusters of computers and make individually each of them less error prone. Hadoop architecture consists of not only Hadoop components but also an amalgamation of different technologies that provides immense capabilities in solving complex business problems, government projects.

4.1 Hadoop YARN

A new component that is included in the Hadoop 2.0 is the YARN (Yet Another Resource Negotiator).The YARN framework is responsible to carry out the Cluster Resource Management. To manage the resources for the Hadoop clusters such as Memory, CPU etc was done with the help of the Cluster Resource Management.

Zookeeper framework provides most of the clusters touse this service to have coordination among themselves and to maintain the shared data with synchronization techniques.

- Nimbus is basically stateless and uses Zookeeper to check the status of the nodes whether they are working or not.
- With the help of this framework Nimbus and Supervisor interact with each other.

5. Hadoop Distributed File System (HDFS)

It is one by which large amount of data is stored into the system. It was developed with the help of distributed system design. It can be run on the commodity hardware. The crucial point for HDFS is that it is highly fault tolerant and the cost for designing the system is not much more. To store and access the data is simple task with it. This large data is stored in different machines. Hadoop has the power to create interface by which interacting with HDFS is possible.

HDFS Architecture

HDFS (Hadoop Distributed File System) has master-slave architecture. HDFS stores huge files running on a cluster of commodity hardware. It works on the principle of storage of less number of large files rather than the huge number of small file It compromises a NameNode and a number of DataNodes.

NameNode: The NameNode is also known as MasterNode. NameNode maintains and manages different DataNodes.

DataNode: DataNode is also known as SlaveNode. It stores actual data in HDFS.

Block: The Related Applications that are compatible in the environment of HDFS are those deal with huge data sets. These applications write their data only one time but they read it one or more than one time and require these reads to be satisfied at streaming speeds. HDFS supports write-once-read-many semantics on files. Each typical block size is used by HDFS is 64 MB and 128 MB. Thus, an HDFS file is chopped up into 64 MB and 128MB chunks, and if possible, each chunk will reside on a various Data Node.

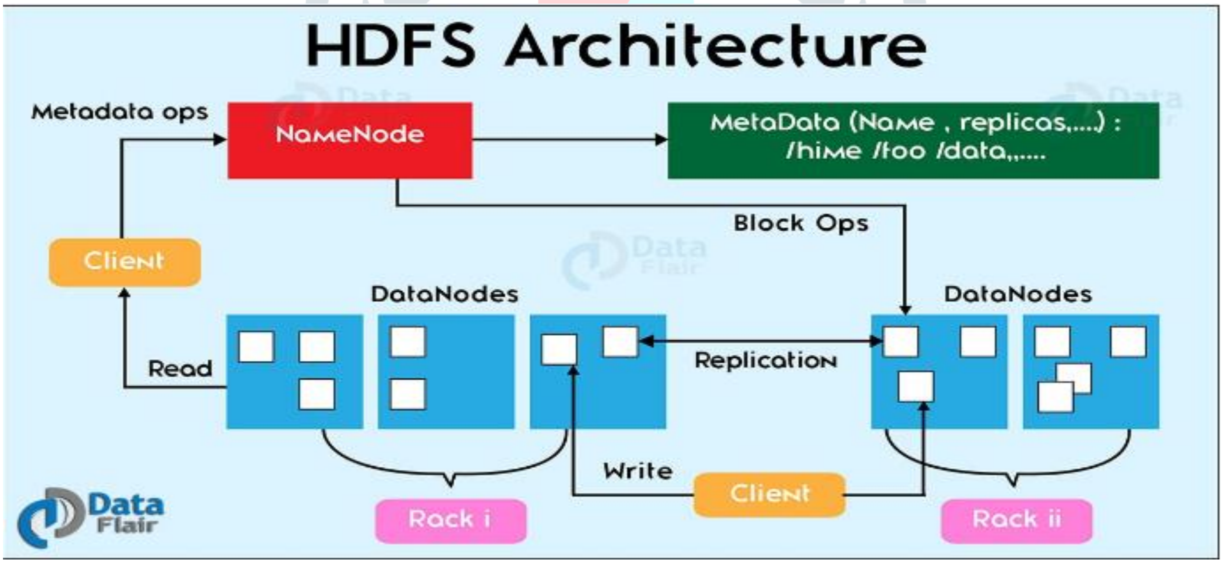

Figure 4: HDFS Architecture Diagram

6. MAPREDUCE

MapReduce isa framework with the help of which we can develop applications to process large data sets in parallel on large clusters. Google developed MapReduce It is a programming model used for distributed computing which is based on java. It is also useful for batch processing of petabytes or terabytes of data stored in the Hadoop.

6.1 Advantages of Using Mapreduce

1. Simple - Applications can be written in any language such as Java, Python etc. by the developer and the Map Reducing Job can run easily .

2. Scalability - Hadoop platform is highly scalable. The business organizations use MapReduce to process petabytes of data that arise from a large number of nodes and can be stored in one cluster of the HDFS.

3. Fast - The problems which used to take lot of time to be processed can now be processed in parallel within few hours with the help of MapReduce.

4. Fault-Tolerant - MapReduce can easily detect the faults and apply fast recovery mechanisms.

6.2 Disadvantages of Using Mapreduce

1. There is no control in the order in which the Maps and Reduce tasks run .

2. MapReduce cannot be applied to a computed value which depends on a previously computed value. Example in Fibonacci series each value depends on the previous values calculated.

7. SECURITY IN HADOOP

The early Hadoop's developers didn't prioritize safety of data, as Hadoop's preliminary cognizance focused on workloads and big datasets were not taken into consideration as sensitive and fall under regulatory challenge. Therefore, it had no model of security and no privacy paradigm, no authentication offers and privacy checks to customers, so anybody could impersonate any other person or inject compiled executable code. In addition, in the distributed environment, all programmers and users possessed equal access privileges to all the datasets within the cluster and any process could be allowed into any datasets within the cluster, and all users could access any dataset (Sedayao, Bhardwaj Gorade, 2014). As only a few safety measures existed in the Hadoop's ecosystem, there have been a lot of miss happening and safety incidents in such an insecure environment. Authentication and authorization of the user was introduced later, but it still had some weaknesses. Later safety inspections dealt with user errors, for example unintended deletion and other non malicious uses (Sharma &Navdeti, 2014; Jam et al. 2014).

With the cutting-edge introduction of Kerberos, and basic ACLs and HDFS permissions, utilization of firewalls, the Hadoop network promotes certain safety features (Valliyappan & Singh (2016).For a Hadoop cluster, Kerberos is not a mandatory requirement, so that it can run entire clusters with any deployment or enforcement protections.

7.1 Hadoop Security Solution

Hadoop is a distributed system which allows us to store huge amounts of data and processing the data in parallel. Hadoop is used as a multi-tenant service and stores sensitive data such as personally identifiable information or financial data. Other organizations, including financial organizations, using Hadoop are beginning to store sensitive data on Hadoop clusters. As a result, strong authentication and authorization is necessary (Lakhe, 2014). The Hadoop ecosystem consists of various components. We need to secure all the other Hadoop ecosystem components.

8. CONCLUSION

We all know that big data represents a significant constraint in today's technological era, and it is a major issue providing security for big data. Almost any large company continues to keep its big data and seems to be prepared to use Hadoop for both its storage and processing. In this paper we tried to discuss about components of Hadoop application, its benefits, benefits of MapReduce and the Hadoop’s Distributed File System and the safety challenges of safeguarding the data and the Ecosystem of Hadoop. Hadoop is analyzed for process and components. HDFS guarantees data integrity across the cluster, including the maintenance of transaction logs. The present article also discusses the MapReduce framework, which incorporates various features for analysis, sorting and processing of the Hadoop stored big data. Major security concerns related to big data were evaluated as well in this paper.

REFERENCES:

1. Huang, J., Nicol, D. M., & Campbell, R. H. (2014, June). Denial-of-service threat to Hadoop/YARN clusters with multi-tenancy. In 2014 IEEE International Congress on Big Data(pp. 48-55).

2. Valliyappan, V., & Singh, P. (2016). Hap: Protecting the apache hadoop clusters with hadoop authentication process using kerberos. In Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics, Springer, New Delhi 151-161.

3. Lee, Y., & Lee, Y. (2013). Toward scalable internet traffic measurement and analysis with hadoop. ACM SIGCOMM Computer Communication Review, 43(1), 5-13.

4. Zaharia, M., Xin, R. S., Wendell, P., Das, T., Armbrust, M., Dave, A., ...&Ghodsi, A. (2016). Apache spark: a unified engine for big data processing. Communications of the ACM, 59(11), 56-65.

5. Demchenko, Y., Ngo, C., &Membrey, P. (2013). Architecture framework and components for the big data ecosystem. Journal of System and Network Engineering, 1-31.

6. Hahanov, V., Gharibi, W., Litvinova, E., &Chumachenko, S. (2015, June). Big data driven cyber analytic system. In 2015 IEEE International Congress on Big Data (pp. 615-622). IEEE.

7. Pääkkönen, P., &Pakkala, D. (2015). Reference architecture and classification of technologies, products and services for big data systems. Big data research, 2(4), 166-186.

8. Sedayao, J., Bhardwaj, R., &Gorade, N. (2014). Making big data, privacy, and anonymization work together in the enterprise: experiences and issues. In 2014 IEEE International Congress on Big Data, pp. 601-607.

9. Sharma, P. P., & Navdeti, C. P. (2014). Securing big data hadoop: a review of security issues, threats and solution. Int. J. Comput. Sci. Inf. Technol, 5(2), 2126-2131.

10. Jam, M. R., Khanli, L. M., Javan, M. S., & Akbari, M. K. (2014, October). A survey on security of Hadoop. In 2014 4th International Conference on Computer and Knowledge Engineering (ICCKE), pp. 716-721.

11. Lakhe, B. (2014). Implementing Granular Authorization. In Practical Hadoop Security (pp. 75-93). Apress, Berkeley, CA.