International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2019 IJSRCSEIT | Volume 5 | Issue 1 | ISSN : 2456-3307
DOI : https://doi.org/10.32628/CSEIT195172

A Review on Hadoop Eco System for Big Data

Anushree Raj[1], Rio D'Souza[2]
[1]M.Sc. Big Data Analytics Department, St Agnes College (Autonomous) Mangalore, Karnataka, India
[2]Computer Science and Engineering Department, St Joseph Engineering College Mangalore, Karnataka, India

ABSTRACT

In this era of information age, a huge amount of data generates every moment through various sources. This enormous data is beyond the processing capability of traditional data management system to manage and analyse the data in a specified time span. This huge amount of data refers to Big Data. Big Data faces numerous challenges in various operations on data such as capturing data, data analysis, data searching, data sharing, data filtering etc. HADOOP has showed a big way of various enterprises for big data management. Big data hadoop deals with the implementation of various industry use cases. To master the Apache Hadoop, we need to understand the hadoop eco system and hadoop architecture. In this paper we brief on the Hadoop architecture and hadoop eco system.

Keywords: Big Data, Hadoop architecture, Hadoop eco system components, HDFS, MapReduce.

I. INTRODUCTION

Hadoop is a open source tool. It provides an efficient framework for running jobs on multiple nodes of clusters. Cluster means a group of systems connected via LAN. Apache Hadoop provides parallel processing of data as it works on multiple machines simultaneously [7]. Big Data and Hadoop efficiently processes large volumes of data on a cluster of commodity hardware [8]. Hadoop is for processing huge volume of data. Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single server to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high availability, the library itself is designed to depict and handle failures at the application layer, so delivering a highly available service on the top of a cluster of computers, each of which may be prone to failures.

The proposed work the second sections gives you a detailed review on Big Data, third section speaks of Hadoop Ecosystem and explains all of its components briefly and the fourth section concludes the paper by providing the main advantages of Hadoop.

II. BIG DATA

Bigdata is a term used to describe a collection of data that is huge in size and yet growing exponentially with time [1]. Big Data generation includes stock exchanges, social media sites, jet engines, etc. Big Data could be Structured, Unstructured or Semi-structured. Volume, Variety, Velocity, and Variability are few characteristics of Big data [2]. It plays a major in Improved customer service, better operational efficiency, Better Decision Making.

There were two major challenges with Big Data [9]:

- ***Big Data Storage:*** To store Big Data, in a flexible infrastructure that scales up in a cost effective manner, was critical.
- ***Big Data Processing:*** Even if a part of Big Data is Stored, processing it would take years.

To solve the storage issue and processing issue, two core components were created in Hadoop – **HDFS** and **YARN**. HDFS solved the storage issue as it stores the data in a distributed fashion and is easily scalable [8]. YARN solved the processing issue by reducing the processing time drastically.

III. HADOOP ECHO SYSTEMS

Hadoop Eco System is aligned in four different layers such as Data Storage, Data processing, Data Access and Data Management. Each layer comprises of different Hadoop eco system components [7]. The figure shows the Hadoop eco system with its components.

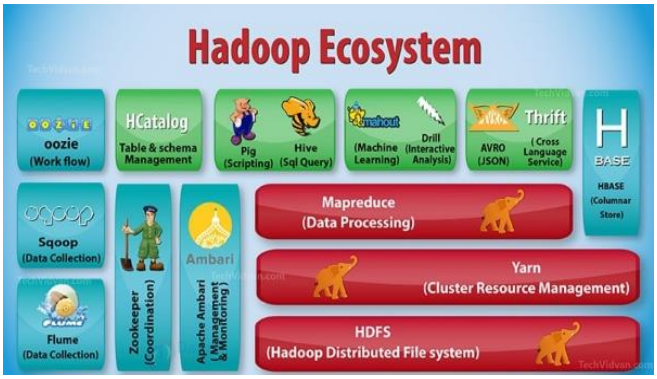

Figure Hadoop Eco Systems

1. Data Storage Layer in Hadoop

Hadoop aims to execute the distributed processing with the minimum latency possible. this is achieved by executing Map processing on the node that stores the data, a concept known as data locality.

The data storage components of hadoop eco system are the HDFS and HBASE.

HDFS:

HDFS is a distributed file system. It supports a single file system name space, which stores data in a traditional hierarchical format of directories and files. Across an instance, data is divided into 64MB chunks that are triple-mirrored across the cluster to provide resiliency. It is optimized for streaming access of large files that are in the 100s of MB upwards on HDFS and access them through MapReduce to process them in batch mode [4]. HDFS files are write once files. There is no concept of random writes. HDFS doesn't do random reads.

HBase:

Apache HBase is a database that stores it's data in a distributed filesystem. The filesystem of choice typically is HDFS owing to the tight integration between HBase and HDFS. HBase provides low latency access to small amounts of data from within a large data set. We can access single rows quickly from a billion row table. It provides flexible data model to work with and data is indexed by the row key. It can perform fast scans across tables [5]. It scales in terms of writes as well as total volume of data.

2. Data Processing Layer in Hadoop

The data processing framework is a tool used to work with the data itself.

The data processing components of hadoop eco system are MapReduce and YARN.

MapReduce:

MapReduce is the tool that actually gets data processed. MapReduce runs as a series of jobs, with each jobs essentially a separate java application that gives out into the date and starts pulling out information as needed [6]. Hadoop is not really a database. It stores data and we can pull data out of it, but there are no queries involved. Hadoop is more a data warehousing system so it needs a system like MapReduce to actually process the data.

YARN:

YARN stands for "Yet Another Resource Negotiator". Yarn runs a non-MapReduce jobs within the Hadoop framework. YARN is a key element of the Hadoop data processing architecture that provides different data handling mechanisms, including interactive SQL and batch processing [3]. 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 allows different data processing methods like graph processing, interactive processing, stream processing as well as batch processing to run and process data stored in HDFS. Therefore YARN opens up Hadoop to other types of distributed applications beyond MapReduce [5]. YARN architecture consists of Resource management, Node management, Application master and container.

3. Data Access Layer in Hadoop

Hadoop can usually hold terabytes or petabytes of data to process, hence Data Access is an extremely important aspect in any project or product, especially with hadoop.

The Data access components of hadoop eco system are:

Hive:

Apache Hive is a component of Hortonworks Data Platform (HDP). Hive provides a SQL-like interface to data stored in HDP [10]. Hive gives a SQL-like interface to query data stored in various databases and file systems that integrate with Hadoop. Hive provides the necessary SQL abstraction to integrate SQL-like queries (HiveQL) into the underlying Java without the need to implement queries in the low-level Java API. Since most data warehousing applications work with SQL-based querying languages, Hive aids portability of SQL-based applications to Hadoop

Pig:

Apache Pig is a platform for managing large sets of data which consists of high-level programming to analyze the data. Pig also consists of the infrastructure to evaluate the programs [7]. The advantages of Pig programming is that it can easily handle parallel processes for managing very large amounts of data. The programming on this platform is basically done using the textual language Pig Latin.

Mahout:

Apache Mahout is an open source library of scalable machine learning algorithms that focuses on clustering, classification, and recommendations [3]. The primitive features of Apache Mahout are: The algorithms of Mahout are written on top of Hadoop, so it works well in distributed environment. Mahout uses the Apache Hadoop library to scale effectively in the cloud. Mahout offers the coder a ready-to-use framework for doing data mining tasks on large volumes of data. Mahout lets applications to analyze large sets of data effectively and in quick time. It Includes several MapReduce enabled clustering implementations such as k-means, fuzzy k-means, Canopy, Dirichlet, and Mean-Shift, Supports Distributed Naive Bayes and Complementary Naive Bayes classification implementations, Comes with distributed fitness function capabilities for evolutionary programming, Includes matrix and vector libraries.

Avro:

Avro is a remote procedure call and data serialization framework developed within Apache's Hadoop project [10]. It is a schema-based serialization utility, which accepts schemas as input. In spite of various schemas being available, Avro follows its own standards of defining schemas like type of file, location of record, name of the record, fields in the record with their corresponding data types.

Sqoop:

Apache Sqoop is a tool that is extensively used to transfer large amounts of data from Hadoop to the relational database servers and vice-versa [7]. Sqoop can be used to import the various types of data from Oracle, MySQL and such other databases. Sqoop needs a connector to connect the different relational databases. Almost all Database vendors make a JDBC connector available specific to that Database, Sqoop needs a JDBC driver of the database for interaction.

Drill:

Apache Drill is a low latency distributed query engine for large-scale datasets, including structured and semi-structured/nested data [3]. Drill is capable of querying nested data in formats like JSON and Parquet and performing dynamic schema discovery. Drill does not require a centralized metadata repository

Thrift:

Thrift is a driver level interface that provides API for client implementation. It combines a software stack with a code generation engine to build cross-platform services which can connect applications written in a variety of languages and frameworks [11].

4. Data Management Layer in Hadoop

Hadoop data management software(entity) is for managing enormous amounts of scattered data. It is part of the ecosystem of modern data management [11].

The Data Management components of hadoop eco system are:

Oozie:

Oozie is a workflow scheduler system to manage Apache Hadoop jobs. Oozie is integrated with the rest of the Hadoop stack supporting several types of Hadoop jobs such as Java MapReduce, Streaming MapReduce, Pig, Hive and Sqoop [10]. Oozie is a scalable, reliable and extensible system. The main Features of Oozie are: It execute and monitor workflows in Hadoop, provides periodic scheduling of workflows, Trigger execution of data availability and uses HTTP and command line interface and web console.

Ambari:

Apache Ambari is an open source administration tool deployed on top of Hadoop cluster and responsible for keeping track of running applications and their status. Apache Ambari can be referred to as an open source web-based management tool that manages, monitors and provisions the health of Hadoop clusters [3]. It provides a highly interactive dashboard which allows the administrators to visualize the progress and status of every application running over the Hadoop cluster.

Its flexible and scalable user-interface allows a range of tools such as Pig, MapReduce, Hive, etc. [11], to be installed on the cluster and administers their performances in a user-friendly fashion.

Flumes:

Apache Flume is a distributed, reliable, and available system for efficiently collecting, aggregating and moving large amounts of log data from many different sources to a centralized data store. Suppose there are 100 servers with unstructured log files [11]. We have to load data in HDFS and analyze the same. We can use flume for this purpose. It has three tiers Agents, Collector ad storage. Flume can capture real time data from the server. It can sink data from logs into HDFS. Flume is used to move the log data generated by application servers into HDFS at a higher speed.

Zookeeper:

Apache Zookeeper is a coordination service for distributed application that enables synchronization across a cluster [3]. Zookeeper in Hadoop can be viewed as centralized repository where distributed applications can put data and get data out of it. It is

used to keep the distributed system functioning together as a single unit, using its synchronization, serialization and coordination goals. For simplicity's sake Zookeeper can be thought of as a file system where we have znodes that store data instead of files or directories storing data. Zookeeper is a Hadoop Admin tool used for managing the jobs in the cluster.

ZooKeeper is a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services. ZooKeeper is simple, distributed, reliable, and fast [11].

- maintaining configuration information: It maintains cluster configuration info which is shared across all the nodes in the cluster.
- naming: Zookeeper can be used as naming service, so that one node in the cluster can find another node in the large cluster ex: 1000 node cluster
- providing distributed synchronization: We can also use zookeeper for solving distributed synchronization problems in the cluster by using locks, Queues etc.
- providing group services: Zookeeper also helps in group service with the selection of a master in the cluster (Leader election process).

ZooKeeper can work in replicated mode as well as standalone mode

HCatalog:

HCatalog is a table storage management tool for Hadoop. It exposes the tabular data of Hive metastore to other Hadoop applications. It enables users with different data processing tools (Pig, MapReduce) to easily write data onto a grid. It ensures that users don't have to worry about where or in what format their data is stored [3].

HCatalog works like a key component of Hive and it enables the users to store their data in any format and any structure.

HCatalog is a table storage management tool for Hadoop that exposes the tabular data of Hive metastore to other Hadoop applications. It enables users with different data processing tools (Pig, MapReduce) to easily write data onto a grid [7]. HCatalog ensures that users don't have to worry about where or in what format their data is stored. This is a small tutorial that explains just the basics of HCatalog and how to use it.

IV. CONCLUSION

Apache Hadoop is the most popular and powerful big data tool, Hadoop provides world's most reliable storage layer – HDFS, a batch Processing engine – MapReduce and a Resource Management Layer – YARN [11]. HADOOP is a highly scalable storage platform, because it can store and distribute very large data sets across hundreds of inexpensive servers that operate in parallel. Unlike traditional relational database systems (RDBMS) that can't scale to process large amounts of data, Hadoop enables businesses to run applications on thousands of nodes involving thousands of terabytes of data. HADOOP also offers a cost-effective storage solution for businesses' exploding data sets. Hadoop enables businesses to easily access new data sources and tap into different types of data (both structured and unstructured) to generate value from that data [13]. This means businesses can use Hadoop to derive valuable business insights from data sources. A key advantage of using Hadoop is its fault tolerance. When data is sent to an individual node, that data is also replicated to other nodes in the cluster, which means that in the event of failure, there is another copy available for use.

V. REFERENCES

[1] 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

[2] 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

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

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

[10] https://intellipaat.com/tutorial/hadooptutorial/introductio n- hadoop/

[11] Apache Hadoop. http://hadoop.apache.org/

[12] Kala Karun. A , Chitharanjan. K," A Review on Hadoop – HDFS Infrastructure Extensions", Conference on Information and Communication Technologies,2013,IEEE

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

Cite this article as :

Anushree Raj, Rio D'Souza, "A Review on Hadoop Eco System for Big Data", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 5 Issue 1, pp. 343-348, January-February 2019. Available at doi : https://doi.org/10.32628/CSEIT195172

Journal URL : http://ijsrcseit.com/CSEIT195172