Volume 4, Issue 6, June-2017, pp. 195-201 ISSN (O): 2349-7084

International Journal of Computer Engineering In Research Trends

Available online at: www.ijcert.org

Survey on Big Data using Apache Hadoop and Spark

Priya Dahiya [1], Chaitra.B [2], Usha Kumari [3]

[1] *Student, Information Science Dept. , Acharya Doctor Sarvepalli Radhakrishnan Rd, Bengaluru, Karnataka 560107, India.*

[2] *Assistant Professor, Information Science Dept. , Acharya Doctor Sarvepalli Radhakrishnan Rd, Bengaluru, Karnataka 560107, India.*

[3] *Assistant Professor, Information Science Dept. , Acharya Doctor Sarvepalli Radhakrishnan Rd, Bengaluru, Karnataka 560107, India.*

Priya.dahiya2905@gmail.com [1], Chaitra@acharya.ac.in [2], Ushakumari@acharya.ac.in [3]

Abstract: Big data is growing rapidly regarding volume, variability, and velocity which make it difficult to process, capture and analyze the data. Hadoop uses MapReduce which has two parts Map and Reduce whereas Spark uses Resilient Distributed Datasets (RDD) and Directed Acyclic Graph (DAG) for processing of large datasets. To store data both of them uses Hadoop Distributed File System (HDFS).This paper shows the architecture and working of Hadoop and Spark and brings out the differences between them and the challenges faced by MapReduce during processing of large datasets and how Spark works on Hadoop YARN.

Keywords: Big data, Spark, Hadoop, HDFS, MapReduce, YARN

---

1. Introduction

Big data is immensely growing with the increase in users on various social networking sites or in industries. Big Data can be classified in the form of structured, unstructured and semi-structured form. Traditional data processing techniques are unable to store and process this huge amount of data, and due to this, they face many challenges in processing Big Data and demands of the end user. Nowadays web traffic, social media content, system data and machine-generated data are growing rapidly.

The five V's of Big Data [1, 2] are velocity, volume, veracity, variability, and variety. Velocity depends on the speed how fast data is growing and processing. Volume determines the size of data whether it can be called as big data or not. Veracity determines the quality of captured and processed data. Variability depends on the inconsistency in data, if data is more inconsistent various techniques are used to manage this. Variety is the type i.e. structured, unstructured and semi-structured and nature of data.

Apache Hadoop is a software framework which is open-source and used to store, manage and process data sets using MapReduce programming model. Apache Hadoop has two parts; HDFS and MapReduce. HDFS is used for storing data in distributed environment. MapReduce process information in the nodes by executing parallelly in the system. MapReduce is popular for its simplicity, scalability, and fault-tolerance. MapReduce is one of the key approaches to meet the demands of computing massive datasets. The MapReduce

challenges are put into three categories: online processing, security and privacy, and data storage.

Apache Spark is a cluster computing framework which is open-source and used for programming clusters with implicit parallelism and fault-tolerance. Apache Spark is used for performing fast and real-time analysis at a lightning fast speed which cannot be done by Hadoop

2. Literature Review

The paper proposed by Ms. Vibhavari Chavan and Prof. Rajesh. N. Pursue [3] describes what is Big data and what are five V's, i.e., velocity, volume, veracity, variability, variety. In this paper, the author has given the working of Hadoop HDFS and MapReduce working. To analyze the huge amount of data author has considered Apache Hadoop as a working model. Data is stored in HDFS and to process a large amount of data it has used the concept of key-value pairs.

Author Ankush Verma etc. [4] has proposed a paper in which gives the difference between Hadoop and Spark. The paper also shows the working of the two, and a comparative study is done on them. Spark has overcome the limitations which are present in the traditional system. Better performance is an important factor for maintaining a massive amount of data.

The working and architecture of Spark when it used along with Hadoop YARN is shown by the authors Wei Huang, Lingkui Meng, Dongying Zhang, and Wen Zhang [5].YARN works in a heterogeneous environment for Apache storm and Tez.

Analyzing and processing large data is quite a tedious job. The traditional systems are not capable enough to handle this large data. Katarina Grolinger, Michael Hayes, etc. [6] proposed a paper which describes briefly about the challenges faced by MapReduce while processing this huge amount of data.

MapReduce. Spark uses (DAG) to divide operators into many stages of tasks and (RDD) for fault tolerance.

This paper focus on the architecture and working of Apache Hadoop and Apache Spark and the challenges faced by MapReduce and differences between Hadoop and Spark. It also shows the operation of Spark on Hadoop YARN and YARN model.

3. Hadoop Architecture

Hadoop stores and compute data by creating clusters using many computers. It can be designed for a single server or thousands of machines to compute and store significant data. It uses MapReduce algorithm to run the application, where the data is processed in parallel with others. In MapReducei.e MR one node acts as a master node and other nodes as slave nodes where master node handles the slave nodes that mean it follows master-slave architecture. Hadoop consists of HDFS and MR.

A.HDFS Layer

Since the data is huge and single node is not capable of storing it, HDFS is used as an alternative. HDFS is based on Google file system (GFS) to store data in multiple nodes by splitting the huge data into smaller parts. The HDFS is mainly designed for storing datasets and

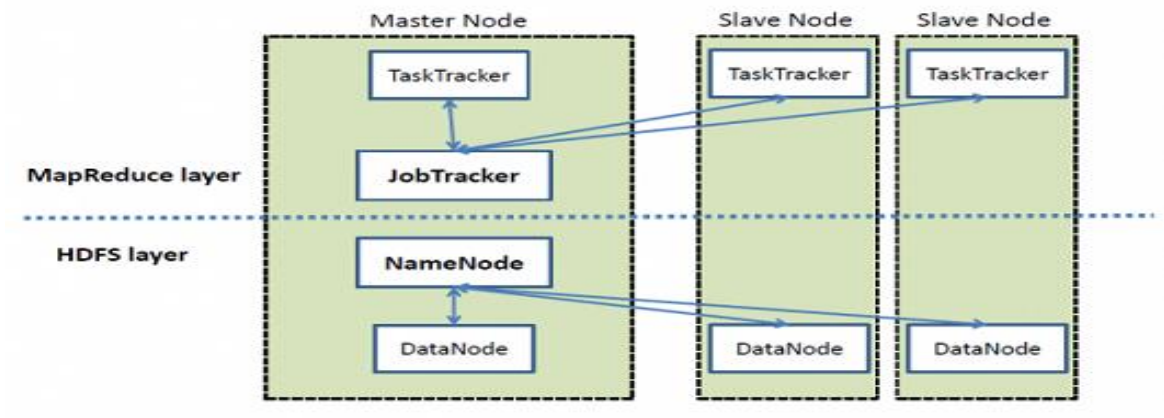

Fig 1.Architecture of Hadoop

To use them for user applications. HDFS has DataNodes which act as a master and NameNode which act as a worker [3]. These nodes are used for performing functions like reading, write, create and delete. NameNode is used for requesting the access permission. If the request is granted, NameNode converts filename into block IDs and multiple DataNodes stores the block and return the list to the client as shown in figure 1.

Name node has information about all DataNodes i.e., data stored by them, which nodes are working actively and which are not i.e. passive nodes, about the free space and job tracker is working efficiently or not. DataNodes stores file location, blocks of data and attributes which are recorded by Name node. Permission to access and modify files is considered as the attributes which are recorded by NameNode. Client node requests NameNode to read the file and location of blocks in HDFS. The NameNode knows the free blocks in the system to use them in future.

The DataNode works by namespace ID and doesn't work if ID is not available and drops the connection with NameNode. New DataNodes have to register with NameNode and receive the namespace ID. Block report is maintained by DataNode and sends this report every hour to NameNode to be up to date. Every ten minutes each DataNode send a signal to check whether it is operating properly or not.

B.MapReduce layer

A large amount of data needs to be properly processed in a distributed environment, and for processing, this data Map Reduce is used. The advantage that MapReduce has is, it makes the workload for handling large data by processing it parallelly on clusters of computers.This makes the system more reliable and fault-tolerant. As shown in fig.1 MapReduce layer has a JobTracker which is used for assigning tasks to the TaskTracker [3]. TaskTracker of slave node gets the tasks to be completed from the Master node JobTracker.

The progress on slave node is monitored by the master node and re-executes tasks when they get failed. Each slave has a TaskTracker for executing tasks assigned by the master. The two-phase used in MapReduce are map phase and reduce phase, and programming language is Java.

4. Spark Architecture

Spark ran on top of Hadoop and used for streaming of data which is in real-time. Spark supports machine learning, SQL queries, graph data processing and streaming data for analysis of big data [4]. Since traditional MapReduce failed to work properly for real-time data, Spark is used as an alternative[8].

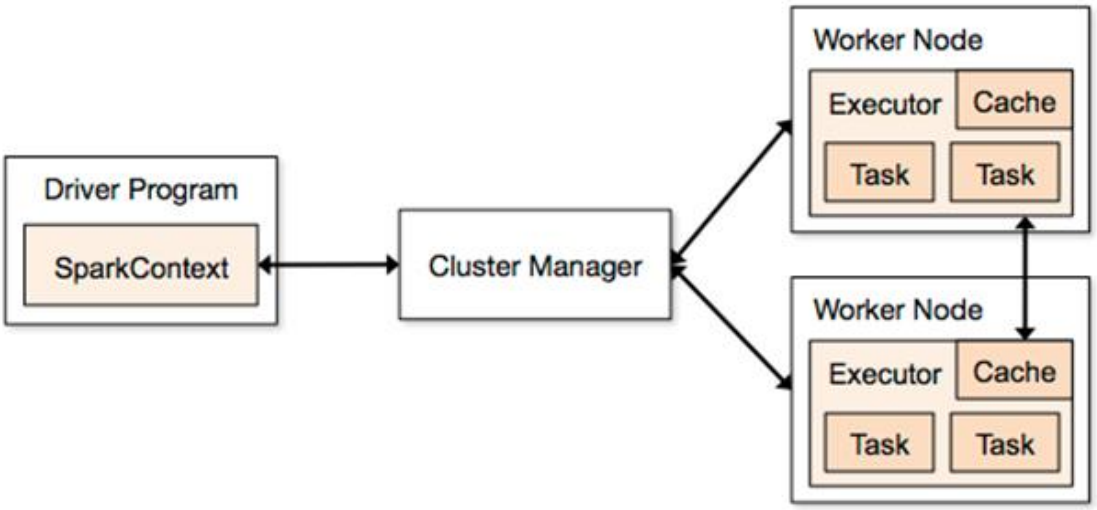

Fig 2. Architecture of Spark

Spark supports languages such as Java and Python, and it is implemented in Scala, and it runs in Java Virtual Machine (JVM). Spark consists of cluster manager, driver program (spark context), executor or

worker and HDFS as shown in fig 2. In spark, a Driver program is considered as the main program. Spark Context is for the coordination of the applications which run on clusters as a set of processes. Processes used for applications are assigned uniquely i.e. they all have their processes and due to this tasks run in multiple threads, and they must have connectivity to worker nodes. These worker nodes run computations and store the data. Programming is written in java or python language which is sent to the executor, and it runs the tasks[8].

The two main key concepts used in Apache Spark are Resilient Distributed Datasets (RDD) and Directed Acyclic Graph (DAG).

A. Resilient Distributed Dataset (RDD)

Resilient Distributed Datasets works as a collection of elements which are operated in parallel. In the distributed file system Spark runs on Hadoop cluster, and RDD is created from files in the format of text or sequence files. RDD is used for reading the objects in the collection and when some partition is lost, it can be rebuilt because RDDs are distributed across a set of machines. The two operations supported by RDD: Transformations and Actions [4].

1. Transformation

A new RDD is returned when transformations are applied. A new RDD is returned when the transformation is applied on the existing RDD and execution is done when an action takes place. Map, flatMap, aggregateByKey, filter, coalesce,reduceByKey and pipe and groupByKey are the examples of some functions of transformation.

2. Action

Action operation such as collect, reduce,countByKey are applied on RDD, and after that, a new value is returned. Action operations are used for evaluation of RDD's. It is used for computation of data processing queries and returns the value.

B. Directed Acyclic Graph

The data flow is cyclic in Directed Acyclic Graph (DAG) engine. To perform on cluster, a DAG of task stages is created by each Spark job. In map and reduce stage DAG is created [4]. It just takes one single stage to complete simple jobs and multiple stages to complete complex jobs in one single run. Thus, jobs completed faster than MapReduce.

5. Difference Between Spark And Hadoop Mapreduce

The difference between Spark and Hadoop MapReduce [7] is shown in the table below:

HADOOP MAPREDUCE	SPARK
The data is stored in the disc.	The data is stored in-memory.
Computing is based on the disc.	Computing relies on RAM.
Fault tolerance is done through replication.	Fault tolerance is done through RDD.
Hard to work with real-time data.	Easy to work with real-time data.
Less costly in comparison to spark.	More costly.
For batch processing only.	Supports interactive query.

Table 1: Hadoop MapReduce vs. Spark

6. Using Spark On Hadoop YARN

For storing and processing data, Hadoop is used in a distributed environment. The commonly used big data platform is Hadoop. Hadoop YARN is the standard operating platform for big data and architecture of Spark YARN [5] is shown in fig. 3. The main component of YARN is ResourceManager, and it allocates and manages containers. Containers represent memory resources and virtual cores. Node manager shows the container status and recycles them dynamically when needed in a distributed environment.

Hadoop YARN has the heterogeneous cluster environment for the framework of Apache Storm and Apache Tez. Task scheduling of these application frameworks is handled by YARN which sends these to each application master (AM).Tasks are being scheduled to the allocated containers by AM. This shows an increase in scalability. To process a huge volume of data AM uses a large number of containers and capacity of these containers is limited to schedule number of tasks. Scaling limitations can be overcome in the cloud. Scheduling of tasks can be done based on data locality.

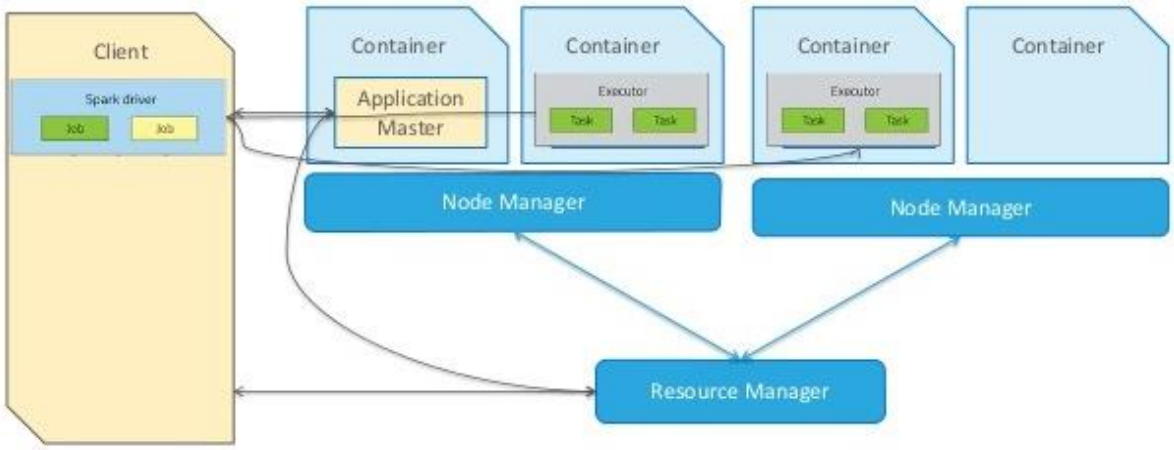

Fig 3.Architecture of Spark YARN model

A. Spark on the YARN Model

How Spark works on YARN model is shown in this. Clients submit an application to YARN platform and Spark master instance request YARN ResourceManager to supply containers [5].Spark Master who runs all the application is used for scheduling tasks in Spark executors that run on the containers. The first process loaded into a container is Spark master. Spark master instance allocates the containers dynamically to Spark executors. Requirements of drivers and executors are provided by clients. According to the requirements of customer YARN platform provides containers for a limited period. Spark master is used to creating RDDs from HDFS, and Spark executors execute the functions of RDD which are being mapped to the parallel tasks. Scheduling of tasks and common result management is done by Spark master. To speed up the transformations in the system RDDs partitions are cached in-memory[9]

7. Challenges In Mapreduce

Challenges in MapReduce include:

A. Data storage

The most used storage system for storing structured data is Relational database management systems (RDBMSs), and it uses SQL for accessing data [6]. Performance, scalability, and availability are the challenges which are faced by RDBMS. MapReduce provides scalability and uses distributed file system such as HDFS. An alternative to Big Data storage is NoSQL and NewSQL. When a number of machines is more, it is important to provide flexibility and scaling, and NoSQL is used for this. To increase the scalability of reading operations, NoSQL works in the same manner as MapReduce. MapReduce works with both semi-structured and unstructured data. Standard MapReduce has poor performance as compared to relational databases. MapReduce and data storage face challenge while handling standardized SQL-like language

B. Online processing

Online processing requires data which is either real-time or quasi-real-time and then process it [6]. MapReduce faces these problems while handling real-time data:

- The input of MapReduce is taken in the form of snapshots of data which are stored in files and during processing contents does not change which makes difficult to handle data because unbounded inputs of data are generated.
- The implementations of MapReduce stores data in distributed and high-overhead file system due to which it face latency in the processing pipeline.

C. Security and privacy

Accounting and auditing are the security issue faced by MapReduce. When the action is performed by someone and to make them responsible for that action is called as accounting, and it is tracked through auditing. Accountability issues are even faced by mappers and reducers while performing tasks. This issue can be overcome by creating accountable MapReduce. In real-time mappers and reducers are tested by auditors to check accountability [6]. To detect malicious mappers or reducers monitoring is done. Proving access control is another security challenge. Since for completing a task multiple access requirements is needed, Velocity is used to determine the access control within limited or fixed time limit. When dealing with huge amount of data privacy becomes the most important parameter. Privacy protection can be done using access control of their information.

8. Conclusion

As data is growing rapidly, it is important to handle this huge amount of data with proper techniques. In this paper, a comparison between Apache Hadoop MapReduce and Apache Spark is shown. They both store data in HDFS, but processing is different in both the cases. The architecture of Hadoop includes HDFS and MapReduce whereas in Spark RDD and DAG is used. Computing is fast in Spark because data in stored in-memory and can be worked with real-time data. Spark can work on Hadoop YARN model. Various challenges faced by MapReduce such as Data storage, online processing and security and privacy Accounting and auditing is mainly concerned for safety and privacy due to the generation of huge amount of data in real-time. Spark is used as an alternative to MapReduce when faster results are needed for real-time data.

References

1. Varsha B.Bobade, "Survey Paper on Big Data and Hadoop", International Research Journal of Engineering and Technology (IRJET) , Volume: 03 Issue: 01 | Jan-2016, e-ISSN: 2395-0056 p-ISSN: 2395-0072.

2. S. Justin Samuel, Koundinya RVP, Kotha Sashidhar and C.R. Bharathi, "A SURVEY ON BIG DATA AND ITS RESEARCH CHALLENGES", VOL. 10, NO. 8, MAY 2015 ISSN 1819-6608, ARPN Journal of Engineering and Applied Sciences.

3.Ms. Vibhavari Chavan, Prof. Rajesh. N. Pursue "Survey Paper On Big Data", Vibhavari
Chavan et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (6) , 2014, 7932-7939.

4. Ankush Verma ,Ashik Hussain Mansuri ,Dr. Neelesh Jain "Big Data Management Processing with Hadoop MapReduce and Spark Technology: A Comparison" 2016 Symposium on Colossal Data Analysis and Networking (CDAN) , 978-1- 5090-0669-4/16/$31.00 © 2016 IEEE.

5. Wei Huang, Lingkui Meng, Dongying Zhang, and Wen Zhang, "In-Memory Parallel Processing of Massive Remotely Sensed Data Using an Apache Spark on Hadoop YARN Model" , IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING, VOL. 10, NO. 1, DECEMBER 2016.

6. Katarina Grolinger, Michael Hayes, Wilson A. Higashino, Alexandra L'Heureux1 David S.Allison ,Miriam A.M. Capretz, "Challenges for MapReduce in Big Data ", 978-1- 4799-5069-0/14 $31.00©2014IEEEDOI10.1109/SERVICES.2014.4.

7. Xiuqin LIN, Peng WANG, Bin WU, "LOG ANALYSIS IN CLOUD COMPUTING ENVIRONMENT WITH HADOOP AND SPARK", 978-1-4799-0094-7/13/$31.00©2013

8. K..Naga Maha Lakshmi et al., International Journal of Computer Engineering In Research Trends ,Volume 3, Issue 3, March-2016, pp. 134-142.

9. Sunil B. Mane et.al, "Product Rating using Opinion Mining", International Journal of Computer Engineering In Research Trends, 4(5):161-168 ,May -2017.

