A REVIEW PAPER ON BIG DATA TECHNOLOGIES AND PLATFORMS

[1]Jyoti Khandelwal, [2]Manoj kumar
[1]Assistant professor, [2]Assistant professor

Department of Computer Science Engineering
Rajasthan Insitute Of Engineering and Technology,Bhankrota,Jaipur,India

Abstract - There is no hard and fast rule about exactly what size a database needs to be in order for the data inside of it to be considered "big." Instead, what typically defines big data is the need for new techniques and tools in order to be able to process it. The large data is analysed using different types of software's. This paper presents an overview of big data's content, methods, advantages followed by discussion on data analysing technologies like Apache Spark.

INTRODUCTION

Big data is an term that describes any Large amount of structured and unstructured data that has the potential to be mined for information. In the 2000's Doug Laney articulated Big Data with 4V's which are widely accepted by users. The three V's are:

Velocity: velocity refers to the speed of data which is streaming. The speed of that data is unpredictable and must deal with time. RFID tags, sensors and smart metering are driving the need to deal with torrents of data in near real time.

Volume: Social networking is increase the depth and breadth of data available about a transaction. There are many other sources which are generating huge amount of data everyday Granter says that everyday these sources are generating 2.5 billion of Gigabyte data.

Variety: The source of data are many like the social media, business transaction, machine to machine data, sensors so the data must be present in text, numerical, audio or video format [1].

I. APACHE SPARK

Spark is an open source cluster computing (distributed computing) framework. Distributed computing is a field of computer science that studies distributed systems. A distributed system is a model in which components located on networked computers communicate and coordinate their actions by passing messages. The components interact with each other in order to achieve a common goal. Spark provides an interface for programing for cluster with implicit data parallelism and fault tolerance. Data parallelism is a form of parallelization of commuting across multiple process it basically focus on distributing the data across different parallel nodes. Data parallelism can be achieved when in multiprocessor system each processor performs the same task on different pieces of distributed data.

Fault tolerance is the property that stops a system to continue operating properly because of some component failure.

Apache spark provides programmers with an application programing interface (API) in computer programing API is a set of routine, protocols and tools for building software and application. And this API is cantered on the data structure which is called as resilient distribute dataset (RDD). RDD basically provide abstraction which is a collection of element dived across the nodes and can be operated parallel and it also control the fault tolerance.

Feature of Apache Spark

1. **Streaming data:** Analysing the data in real time is very hard task to do apache spark provide this facility to analyse the streaming data live.
2. **Machine learning:** Machine learning is a process in which machine learn from various types of data and act on it.
3. **High Speed:** Analysing data in real time is very hard and it could be very slow in process but Spark has a feature of high speed it can analyse the data 100 time faster than Apache Hadoop.[2]

A. Implementation

Apache Spark implementation has three major components: Resilient Distributed datasets, Iterative Operation and Transformation of resilient distributed dataset.

Resilient Distributed Datasets (RDD) is a fundamental data structure of Spark. It is an immutable distributed collection of objects. Each dataset in RDD is divided into logical partitions, which may be computed on different nodes of the cluster. RDDs can contain any type of Python, Java, or Scale objects, including user-defined classes. Formally, an RDD is a read-only, partitioned collection of records. RDDs can be created through deterministic operations on either data on stable storage or other RDDs.

Iteration on RDD has multiple phases first the live stream data come to Hadoop and then it come to spark so in first phase known as iteration-1 in this map reduce function run on the live data and send it to distributed memory then the second Iteration-2 again run on the data which come from distributed system and there are many phase here which produce the output at last.

RDD transformations returns pointer to new RDD and allow you to create dependencies between RDDs. Each RDD in chain (String of Dependencies) has a functioning for calculating its data and has a source point (dependency) to its parent RDD.So, RDD transformation is not a set of data but the only step telling Spark how to get data and what to do with it.

B. MAPR

Simplified Data Processing on Large Clusters

MapReduce is a kind of framework which helps us to write easy application which processes a very big amount of data in real time. MapReduce works on Large Cluster in Parallel which means using MapReduce we are able to process multi-terabyte of data and all this process is done in fault tolerant manner. MapReduce basically are two functionality one is "MAP" and another one is "Reduce". Map functionality basically divides the data into multiple parts of dataset and these data set are called as tuples which is a combination of key and value. Once Map done its work then Reduce functionality come in by the name we can understand that this functionality of reduce is to eliminate the poor data and get the meaningful data as an output. The Reduce task always works on input and this input is generated by the map functionality as an output. In present day there are hundreds of MapReduce programs are implemented and these programs are executed everyday on Google clusters.

C. Programming model

Generally MapReduce paradigm is depend 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: The map or mapper's job is to process the input data. The input data is generally in the form of file or directory and it is stored in the Hadoop file system (HDFS). 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 a combination of the Shuffle stage and Reduce stage the reduce stage works is to take input from maps stage output and start shuffling the data as per the given job.so what it do is it fetch the data from the HDFS and during the MapReduce job Hadoop sends these both jobs to server the Hadoop framework basically handles all the details like data passing, issuing task, verifying task completion and copying the data from one cluster to another in this technique most of the computing task take place on nodes with data on local disk which help to reduce the network traffic.

Once the task is completed it gathers all the reduced data from clusters and reduces the data to form an appropriate result. For example an user can find out a particular word from a novel on a single server but this process would be very time consuming but if this task is divide among let's say 26 peoples it means each person has one paper and they will find out that particular word so this process of splitting the data is called as map once all 26 persons find out the words then user will gather all the data and merge them once user have that data so they can find out that how many time that word occur in novel so gathering the data and producing the appropriate output is done by reduce job and if a person leave job in between the process at the same time another person takes the job and complete it this characteristics called as Fault Tolerance of system.[3]

II. HADOOP

Distributing storage and distributing processing use Apache Hadoop as a open source framework of very large set of data on computer cluster. Computer cluster is a collection of many loosely and tightly connection of computer which work as a single computer machine. According to the traditional approach, the data was stored in a RDBMS like MYSQL or oracle database, processed and sent to the user for analysis purposes. The problem came when the amount of data became too large to be accommodated by standard database servers.
every day 2.5 billon Gigabyte data is generated so it cannot be store in traditional database so in Hadoop we use NOSQL instead of SQL. NOSQL basically is lightweight Sql. Hadoop don't support Structured data it support Semi and unstructured data so we use NOSQL like HIVE. Hive is a data warehouse where it can store 2.5 billon of Gigabyte every day and much more and to do manipulation of data we use Hive query language.

There is one more thing in Hadoop. It works on write once and read many. Means data can be read many time but we can only write the data one time unlike in traditional RDBMS data can be manipulated using Update command but in Hadoop we don't have this privilege.[4]

III. BENEFITS

Some of the reasons behind using Hadoop in organization is its' power to store, manage and analyse large amounts of structured and unstructured data there are following benefits using Hadoop which are:

1. Scalability and Performance
2. Reliability
3. Flexibility
4. Low Cost[5]

IV. HADOOPDB(HDFS)

HDFS is basically design to collect the big data which are coming from many data generator like Facebook Yahoo etc. in a short time and it distribute this data on different-different nodes.[6] HDFS is able to in writing the program handling their allocation, processing data and generating the final outcome. The key component of HDFS is:

- Name node.
- Data node.
- Job tracker.
- Task tracker

- **Name node:** Name node in Hadoop is the node where Hadoop stores the all location of files in HDFS. In others words it holds the all information about file and HDFS. Whenever a new file stores in the HDFS all the information related to that file like its location is track by Name node.

- **Data node:** The work of data node is to store the files in HDFS. It manages the file block within the node. Data node sends information to Name node about the files and block which are store in that node and respond to Name node for all file related operations.

- **Job tracker:** Job Tracker is responsible for taking in requests from a client and assigning Task Trackers with tasks to be performed. The Job Tracker tries to assign tasks to the Task Tracker on the Data Node where the data is locally present (Data Locality). If that is not possible it will at least try to assign tasks to Task Trackers within the same rack. If for some reason the node fails the Job Tracker assigns the task to another Task Tracker where the replica of the data exists since the data blocks are replicated across the Data Nodes.
- **Task Tracker:** Task Tracker is a daemon that accepts tasks (Map, Reduce and Shuffle) from the Job Tracker. The Task Tracker keeps sending a heartbeat message to the Job Tracker to notify that it is alive. Along with the heartbeat it also notify to tasks about the free slots. Task Tracker starts and monitors the Map & Reduce Tasks and sends progress/status information back to the Job Tracker.[6]

V. PERFORMANCE COMPARISON

We cannot use only apache spark for analyse of big data because big data need to be distribute which cannot be done by Spark so for that Apache Hadoop and spark work together where Hadoop distribute the files by dividing it in 128MB blocks but in terms of performance Spark has Advantage because it can analyse the big data in live stream and it can perform operation on data 100 time faster than Hadoop.

VI. CONCLUSION

We have described Apache Spark, which basically analyse the data faster than any other framework we have found that there are significant advantages of using Apache Spark.

The MapReduce programming model has been developed and used by Google in many different fields. We attribute this success to several reasons. First, the model of MapR is very easy to use, even for programmers who don't have experience with parallel and distributed systems, A large variety of problems are easily delivered as MapReduce computations. For example, MapReduce is utilized for peer group of data for Google production and web services.it is also used for data Mining, sorting and for machine learning. Another use of MapReduce is fetching the data from thousands of clusters which are relevant and accurate. The implementation uses the machine resources very efficiently and analyse the data.

Furthermore Hadoop and hive are very good open source project. We believe that future version of Hadoop can enhance the performance. Hadoop HDFS has a very good option rather than using any other NoSql because in hdfs we have nodes which can store the data as well as it can track the operation simultaneously. The capacity of HDFS to directly include Hadoop makes the analysis flexible and stretchable for performing operations at large scales expected of future workload.

REFERENCES

[1] Tushar Vyas, Pankaj Chittora, "Big data and Future Examination system", 2015.
[2] Rahul Suresh, "Apache Spark and the future of big data analytics", https://suyati.com/blog/apache-spark-and-the-future-of-big-data-analytics,2015
[3] Dr.Urmila R.Pol "Big Data Analysis: Comparison of Hadoop MapReduce and ApacheSpark",2016
[4] Harshawardhan S.bhosale, "A Review Paper on Big Data and Hadoop",2014
[5] Technavio, "Top 5 Benefits Of Using Hadoop", https://www.technavio.com/blog/top-5-benefits-using-hadoop,2016
[6] Steven Haines, "Big Data Analysis with MapReduce and Hadoop", http://www.informit.com/articles/article.aspx?p=2008905,2013
[7] Dean Jeffrey et.al "MapReduce: Simplified Data Processing on large Clusters" Google, Inc. OSDI 2004.