Spark – An Efficient Framework for Large Scale Data Analytics

A.C.Priya Ranjani [1] , **Dr. M.Sridhar** [2]

[1] Assistant Professor
Department of Computer Science and Engineering
Vijaya Institute of Technology for Women
Vijayawada, India.
acpranjani@gmail.com

[2] Associate Professor,
Department of Computer Applications
R.V.R & J.C College of Engineering
Guntur, India.
mandapati12@gmail.com

Abstract- Big Data has become a buzz word in recent times. Many organizations are relying on Big Data to forecast the future trends of their businesses and increase their market value. Hadoop and Spark are two popular technologies that can be used to gain deeper analysis of voluminous and heterogeneous data. Although both of these are used for large scale data analytics but their performances vary. Spark is gaining significance as it performs in-memory computations for increased speed and data process over MapReduce. In this paper, the primary components of Hadoop, Spark and the advantages of Spark over Hadoop MapReduce were discussed.

Index Terms– Big Data, MapReduce, Spark, Hadoop

I. INTRODUCTION

Every second, petabytes of data is being generated at a rapid speed from various sources like social media, stock market transactions, sales, web logs, sensors, genomics, web documents, internet images, movies, MP3 files and lot many. This data is categorized as "Big Data" due to its sheer volume, variety, velocity and veracity. Recently other V's such as validity, value, variability, venue, vocabulary and vagueness were also added to explain the complexity of Big Data.

Big Data is not just about lots of data, it provides an opportunity [1] to find new insights into the existing data as well provides guidelines to capture and analyze your future data. With the advent of new technologies a wide range of people including academicians, marketers, governmental organizations, educational institutions, and motivated individuals produce, share, organize and interact with Big Data.

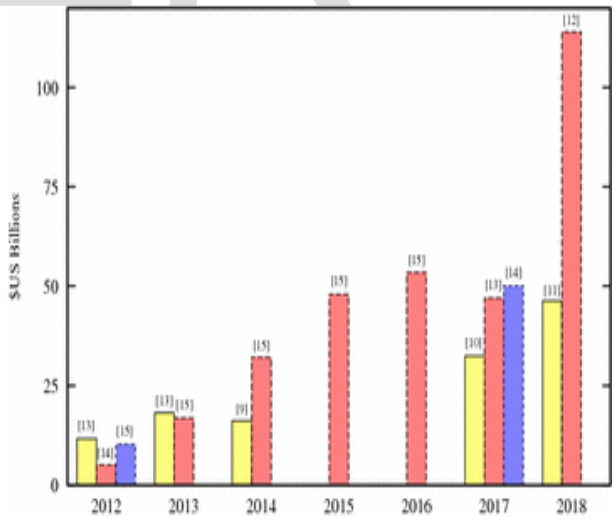

FIGURE 1. GROWTH OF BIG DATA

The report of IDC [2] indicates that the marketing of Big Data is about $16.1 billion in 2014. Another report of IDC [3] forecasts that it will grow upto $32.4 billion by 2017. The reports of [4] and [5] further pointed out that the marketing of Big Data will be $46.34 billion and $114 billion by 2018

respectively. In figure 1 , we can find the rapid growth of Big Data in the coming future.

The rest of the paper is organized as follows. Section II explains the major components of Hadoop framework and limitations of MapReduce. Section III discusses about the Spark technology, its advantages and some applications in which performance gain has been achieved with introduction of Spark and in Section IV concludes the paper.

II HADOOP

Heterogeneity, scaling, performance, timeliness, complexity, cost, and security problems with Big Data impede the progress at all phases of the pipeline that can create value from data. Apache Hadoop [6] is an open source framework that provides solutions for handling Big Data along with extensive processing and analysis. It was created by Doug Cutting in 2005. Hadoop has two major components namely HDFS (Hadoop Distributed File System) [7] and the Map Reduce [8] framework.

A. MapReduce

The programming model used in Hadoop is MapReduce which was proposed by Dean and Ghemawat at Google. MapReduce follows the functional programming model, and performs explicit synchronization across computational stages. MapReduce is the basic data processing scheme used in Hadoop which includes breaking the entire task into two parts, known as mappers and reducers. At a high-level, mappers read the data from HDFS, process it and generate some intermediate results to the reducers. Reducers are used to aggregate the intermediate results to generate the final output which is again written to HDFS. A typical Hadoop job involves running several mappers and reducers across different nodes in the clusters. Users can define map() and reduce() functions.

B. HDFS

The Hadoop Distributed File System (HDFS) is designed to run on commodity hardware. It is highly fault-tolerant and provides high throughput access to large datasets. HDFS has a master/slave architecture [9] . A HDFS cluster consists of a single NameNode which is a master server that manages the file system namespace and regulates access to files by clients. For every node in a cluster, there will be a DataNode which manages the data storage of their system. Internally a file is split into one or more blocks and these blocks are stored in a set of DataNodes. The NameNode executes file system namespace operations like opening, closing, renaming files and directories and also determines the mapping of blocks to DataNodes. The DataNodes are responsible for serving read and write requests from the file system's clients. They perform block creation, deletion, and replication upon instruction from the NameNode.

C. Limitations

One of the major drawbacks [11] of Map Reduce is its inefficiency in running iterative algorithms. Mappers read the same data again and again from the disk. Hence, after each iteration, the results have to be written back to the disk to pass them onto the next iteration. This makes disk access a major bottleneck which significantly degrades the performance. For each iteration, a new mapper and reducer have to be initialized. Regarding storage system, most of the Hadoop applications spend more than 90% of the time doing HDFS read-write operations. Moreover, Hadoop/MapReduce does not support data pipelining or overlap of the Map and the Reduce phases.

III SPARK

Apache Spark [10] is a high performance framework for analyzing large datasets. It was initially developed by Matei Zaharia at UC Berkeley AMPLab in 2009, and open sourced in 2010 as an Apache project. Spark has several advantages compared to other Big Data and MapReduce technologies like Hadoop and Storm. Spark requires a cluster manager and a distributed storage system. For cluster management, Spark supports standalone (native Spark cluster), Hadoop YARN or Apache Mesos. For distributed storage, Spark can interface with a wide variety including HDFS, Cassandra, Amazon S3.

A. Characteristics of Spark

i) Speed: It enables applications in Hadoop clusters to run up to 100 times faster in memory and 10 times faster on disk.
ii) Ease of Use: Applications can be easily written in Java, Scala or Python. It comes with a built-in set of over 80 high-level operators and you can use them interactively to query data within the shell.
iii)Advanced Analytics: Spark not only supports map and reduce functions, it also supports SQL queries, Streaming data, Machine Learning (ML), and Graph algorithms.
*iv)Iterative and Interactive Applications***:** Spark holds intermediate results in memory rather than writing them to disk which is very useful especially when you need to work on the same dataset multiple times. It is designed to be an execution engine that works both in-memory and on-disk. Spark operators perform external operations when data does not fit in memory.
v)In-memory Computations: It is an approach to querying the data when it resides in computer's random access memory (RAM), as opposed to querying data that is stored on physical

disk. Spark supports and decreases the response time to a great extent. It helps business customers to quickly detect patterns, analyze massive data volumes on the fly and perform their operations quickly.

vi)Resilient Distributed Datasets (RDD): The main abstraction of Spark is Resilient Distributed Dataset (RDD), which represents a read-only collection of objects partitioned across a set of machines that can be rebuilt if the partition is lost. Users can explicitly cache an RDD [10] in memory across machines and reuse it in multiple MapReduce-like parallel operations. RDDs are fault tolerant, if a partition of an RDD is lost, it can be rebuilt with the help of other existing RDD's. There are two types of operations that are performed on RDD's:

Transformation: When transformations are applied on RDD's, they return a new RDD and not a single value. Transformations are lazily evaluated; they are executed only when an action runs on it. Some of the transformation functions are map, filter, reduceByKey, flatMap groupByKey, union and intersection.

Action: When Action operations are applied on RDD's, they evaluate and return a single value. All the queries that process the data are computed when an action function is called and return the result. Some Action operations are first, take, reduce, collect, count, foreach and countByKey.

*vii)Directed Acyclic Graph (DAG) :*Spark consists of an advanced Directed Acyclic Graph(DAG) engine which supports cyclic data flow. Each Spark job creates a DAG of task stages to be performed on the cluster. DAGs created by Spark can contain any number of stages, as compared to MapReduce, which creates a DAG with two stages - map and reduce. This allows simple jobs to complete after just one stage, and more complex tasks to complete in a single run of many stages, rather than splitting it into multiple jobs. Thus, jobs complete faster than they would in MapReduce.

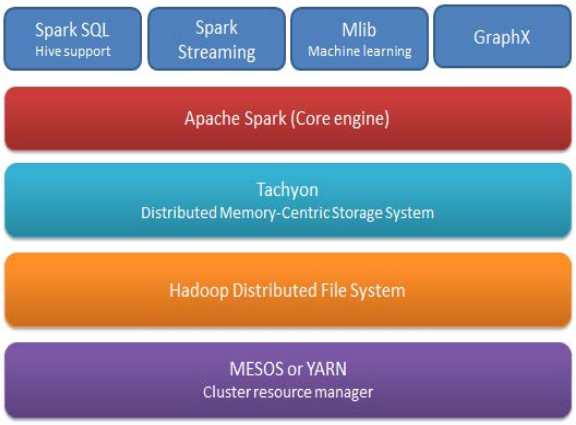

FIGURE 2. SPARK ECOSYSTEM

B. *Spark Ecosystem*

Other than Spark Core API, there are additional libraries [12] that are part of the Spark ecosystem as seen in figure 2. They provide additional capabilities in Big Data analytics and Machine Learning areas. These libraries include:

i)Spark Streaming: It can be used for processing the real-time streaming data. This is based on micro batch style of computing and processing. It uses the DStream which is basically a series of RDDs, to process the real-time data.

ii)Spark SQL: It provides the capability to expose the Spark datasets over JDBC API and allow running the SQL like queries on Spark data using traditional BI and visualization tools. Spark SQL [13] allows the users to ETL their data from different formats it is currently in (like JSON, Parquet, a Database), transform it, and expose it for adhoc querying.

iii)Spark MLlib: MLlib is Spark's scalable machine learning library consisting of common learning algorithms and utilities, including classification, regression, clustering, collaborative filtering, dimensionality reduction, as well as underlying optimization primitives.

iv)Spark GraphX : The GraphX API enables users to view data both as a graph and as collections (RDDs) without data movement or duplication. By incorporating recent advances in graph-parallel systems, GraphX is able to optimize the execution of graph operations. At a high level, GraphX extends the Spark RDD by introducing the Resilient Distributed Property Graph, a directed multi-graph with properties attached to each vertex and edge. To support graph computation, GraphX exposes a set of fundamental operators like subgraph, joinVertices, aggregateMessages etc.

iv)Taychon : It is a fault tolerant distributed file system[14] which enables file sharing at memory-speed across cluster frameworks, such as Spark and MapReduce. It caches working set files in memory, thereby avoiding going to disk to load datasets that are frequently read.

C. *Performance Evaluation of Spark*

Satish & Gopalani [15] compared MapReduce and Spark, providing the performance analysis using a standard machine learning algorithm for clustering using K-Means. They have taken datasets of 64MB, 1240 MB with a single node and 1240MB with two nodes and monitored the performance in terms of the time taken for clustering using K-Means algorithm. The results in tables 1 & 2 clearly showed that the performance of Spark is considerably high in terms of processing time.

TABLE 1 : RESULTS FOR K-MEANS USING SPARK

Dataset Size	Nodes	Time (s)
62MB	1	18

1240MB	1	149
1240MB	2	85

TABLE 2 : RESULTS FOR K-MEANS USING MAPREDUCE

Dataset Size	Nodes	Time (s)
62MB	1	44
1240MB	1	291
1240MB	2	163

Juwei Shi and Yunjie Qiu [16] compared MapReduce and Spark in terms of three architectural components shuffle, execution model , caching and evaluated them through detailed experiments. Workloads like word count, sort K-means, page rank were used in this study.

Apache spark is emerging as the cluster computing platform for Smart Grid [17] Big Data applications. The power systems are very dynamic in nature and disturbances can occur within few milliseconds. The traditional monitoring and measurement infrastructure has the ability to sense data once in 4-6 seconds without any time synchronization. The data collected from these sensors is huge. The Smart-grid can be made more intelligent by processing and deriving new information from the sensor data in real time. Apache spark paves the way to process the PMU (Phasor Measurement Unit) data without much latency to give an accurate view of the grid.

Jongwook Woo [18] empirically found that Market Basket Analysis (MBA) executed on Spark is much faster than legacy MapReduce. The MBA code is written in Scala and processed on AWS EC2. The experimental results in figure 5 show that the more nodes, the better performance is achieved. Besides, the more memory and CPUs, the better performance is gained.

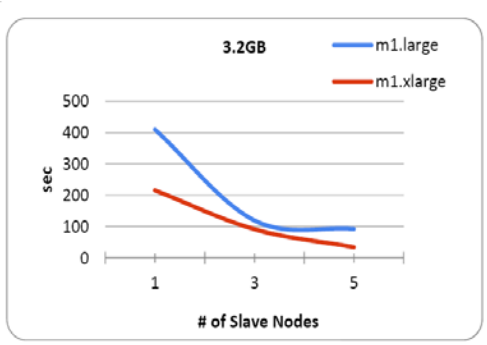

FIGURE 5. PERFORMANCE FOR 3.2GB FILE

IV CONCLUSION

This paper provides an overview of both Hadoop and Spark frameworks and efficiency of Spark over Hadoop . It was found that Spark is a very strong contender due its ability for in-memory computations, interactive querying and stream processing. Although Spark is reported to work up to 100 times faster than Hadoop in certain circumstances, it does not provide its own distributed storage system. So, it requires one provided by a third-party. For this reason many Big Data projects involve installing Spark on top of Hadoop, where Spark's advanced analytics applications can make use of data stored using the Hadoop Distributed File System (HDFS).

REFERENCES

[1] Chun-Wei Tsai , Chin-Feng LaI , Han-Chieh Chao and Athanasios V. Vasilakos ; "Big Data Analytics: a survey ", Journal of Big Data, vol. 2, no. 1, 2015

[2] Press G. $16.1 billion Big Data market: 2014 predictions from IDC and IIA, Forbes, Tech. Rep. 2013.

[3] Big Data and analytics—an IDC four pillar research area, IDC, Tech. Rep. 2013

[4] Taft DK. Big Data market to reach $46.34 billion by 2018, EWEEK, Tech. Rep. 2013.

[5] Research A. Big Data spending to reach $114 billion in 2018; look for machine learning to drive analytics, ABI Research, Tech. Rep. 2013

[6] http://hadoop.apache.org/.

[7] Shvachko K, HairongKuang, Radia S, Chansler R; "The Hadoop Distributed File System "; Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium

[8] Jeffrey Dean and Sanjay Ghemawat; "MapReduce : Simplified data processing on large clusters" ; In OSDI'04: 2004.

[9] HDFS Architecture Guide by Dhruba Borthakur ,2008

[10] Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, Ion Stoica ; "Spark : Cluster Computing with Working Sets "; In HotCloud'10 Proceedings of the 2nd USENIX conference on Hot topics in cloud computing - 2010

[11] VasilikiKalavri, Vladimir Vlassov; "MapReduce: Limitations, Optimizations and Open Issues" ; 11th IEEE International Symposium on Parallel and Distributes Processing - 2013

[12]http://spark.apache.org/docs/latest/programming-guide.html

[13] Michael Armbrust, Reynold S. Xin, Cheng Lian, Yin Huai, Davies Liu, Joseph K. Bradley, Xiangrui Meng, Tomer Kaftan, Michael J. Fran ;" Spark SQL: Relational Data Processing in Spark"; In SIGMOD'15, 2015

[14] Tachyon http://tachyon-project.org

[15] Satish Gopalani , Rohan Arora ; "Comparing Apache Spark and Map Reduce with Performance Analysis using K-Means "; International Journal of Computer Applications , 2015

[16] Juwei Shiz, Yunjie Qiuy, Umar Farooq Minhasx, Limei Jiaoy, Chen Wang, Berthold Reinwaldx, and Fatma O zcanx ; "Clash of the Titans: MapReduce vs. Spark for Large Scale Data Analytics"; Proceedings of the VLDB Endowment ; vol. 8 , no. 13, 2014

[17] Shyam R, Bharathi Ganesh HB, Sachin Kumar S, Prabaharan Poornachandran, Soman KP ; "Apache Spark a Big Data Analytics Platform for Smart Grid "; Procedia Technology, 2015

[18] Jongwook Woo ;"Market Basket Analysis using Spark" ; ARPN Journal of Science and Technology ; vol. 5, no. 4, April 2015