Survey on High Performance Analytics of Bigdata with Apache Spark

Ramkrushna C. Maheshwar
Department of Computer Engineering
Research Scholar at KL University,
Working at I2IT, Pune
Guntur, Andhra Pradesh, India.
remomaheshwar1987@gmail.com

D. Haritha
Professor, Department of Computer Engineering
KL University
Guntur, Andhra Pradesh, India.
haritha_donavalli@kluniversity.in

***Abstract*—This paper lays attention upon the advantages of Apache Spark over Hadoop MapReduce and analysis of real time data using time-series analysis. As Hadoop MapReduce is a widely used and famous execution engine for working with the storage and analysis of large datasets. In MapReduce, the data is read from the disk and the result is written to the Hadoop Distributed File System (HDFS) after a particular iteration and then the data is read from the HDFS for the next iteration. This whole process consumes a lot of disk space and time as well. The users had been objecting the problem of high latency and fault tolerance of the entire system. To overcome the issues and disadvantages of MapReduce, Apache Spark was developed. Apache Spark is an open-source project that ensures lower latency queries, iterative computations and real time processing on similar data. This paper also focuses on time-series analysis in Hadoop and Spark environment which processes and does analysis of real-time data and generates a pattern out of it to get a clearer glimpse of the statistics and characteristics of data thus making Spark even more efficient over MapReduce.**

***Index Terms*— Bigdata Analytics, Apache Spark, Time Series Analysis, HDFS, Hadoop, High Performance Analytics.**

I. INTRODUCTION

The data is emerging from almost every nook and cranny of the world. It requires a great memory size to store such big raw data and it is further very complex to process and analyze this huge amount of data that gets accumulated. Hadoop MapReduce is a framework that processes the large datasets using two functions, namely, map and reduce. Map function takes a set of data and converts into another set of data where each element is then broken down into separate key-value pairs. Reduce function collects output from map function and considering it as an input, it combines the data tuples into smaller set of tuples. Hence in MapReduce, the data is distributed over a cluster and then it is processed. MapReduce has high latency and fault tolerance issues which makes it inappropriate for the users to opt.

Apache Spark was developed to enhance and overcome the functionalities of MapReduce. Spark can be called an advancement of MapReduce. Spark is said to process data 100x faster than MapReduce because it performs in-memory processing of data where there is no time spent in moving the data in and out of the disk. Spark supports streaming of data along with distributed processing. This combination is essential in giving a real-time processing of data, unlike MapReduce. Spark comes with a Resilient Distributed Dataset[3][5], which helps in backtracking and completing a task instead of starting everything from the scratch. Spark is integrated with an in-built library, called Spark MLLib which consists of machine learning algorithm for faster execution of programs. Spark's functional programming is integrated with relational processing using Spark SQL[9], Which is used for optimized storage of queries. For analyzing and bringing out a pattern through the huge amount of data, SparkR[21] is used.

We are clear with the idea that huge amount of data gets aggregated and it is stored, processed and analyzed. We will see another concept, i.e. the Time Series Analysis which is used for analysis of the real-time data and sketching out a meaningful pattern out of that processed data. It gives a meaning to the raw data. For Example, the data of daily weather like temperature, humidity, rainfall etc. is all noted each second and further weather forecast is predicted according to the data and pattern out of the stored data. This is called Time-series Analysis of real-time data. Organization of this paper is divided into four sections. (1) Introduction (2) Literature review for understanding basic concepts. (3) System Model (4) Conclusion.

II. LITERATURE REVIEW

A. R Language

R[1][10][12][22] is a free and open-source implementation of the S statistical programming language and computing environment developed by Ihaka and Gentleman in 1996. It is a language that has become an existing standard among statisticians for the development of statistical software. The main objective was to develop a language that majorly emphasized on delivering a user-friendly and better way to perform data analysis, graphical models and statistics. It is visualization based statistic software that gives scientist control of performing the whole data analysis. R is the most inclusive statistical analysis package available. It is the combination of all of the criterion statistical tests, analyses, and models, as well as providing a comprehensive language for manipulating and managing data. The advantage of having visualized data

is that it can be understood more effectively and efficiently compared to the raw numbers alone, hence making it easier from users perspective. Some of the alluring visualization packages available are ggvis, ggplot2, rCharts and googleV[7] is used for displaying analytics results. In order to improve R's slow performance, multiple packages namely pqR, Riposte, renjin and FastR are used. R being a cross-platform runs on various different hardware and operating system. It is popularly used on Microsoft Windows, GNU/Linux and, Macintosh, running on both 32 and 64 bit processors. Moreover R is even compatible to be used along with many other tools used for importing data, for example SAS, CSV les and SPSS, or directly from Microsoft Access, Microsoft Excel, SQLite, Oracle and MySQL[1]. The various format in which R produce graphics output are JPG, PDF,SVG and PNG format, and tabular format for LATEX and HTML. They also provide package collections for performing different tasks.

Some task views in context with data mining are:

- Machine Learning & Statistical Learning;
- Cluster Analysis & Finite Mixture Models;
- Time Series Analysis;
- Multivariate Statistics; and
- Analysis of Spatial Data.

B. HBase

HBase[8] is a column-oriented database management system that runs on top of HDFS[2]. Many big data use cases takes sparse data sets into consideration hence making HBase a good option. HBase not being similar to relational database systems also does not support SQL in general. HBase applications are written in Python, Java, R and Scala much like a typical Spark based application.

C. Apache Spark

In order to perform general data analytics on Hadoop like distributed computing clusters, frameworks like Apache spark [16][17][20][24] is used. One of the features of Apache Spark is that it provides more optimized memory computation while giving better data process and increased speed over map reduce. While it access hadoop data store (HDFS) and runs on top of existing hadoop cluster, it can also process Streaming data from HDFS, Twitter, Kafka and structured data in Hive. It can be used for fast interactive queries that finish within seconds and real-time stream data processing. Apart from this, the other additional feature in Spark are streaming, ease of use, speed, complex analytics and Combines SQL, also it can run anywhere. Spark allows applications in Hadoop clusters to run up to and 10x faster when running on disk and 100x faster when running in memory. This is possible by reducing the actual number of read/ write operations to and from disc. Spark also gives you the feature to quickly write applications in Scala, Java, or Python. This in return helps developers to create and run their applications in the programming languages they are more familiar with thereby making it easy to build parallel apps. Eventually, Spark also supports streaming data, SQL queries, and complex analytics such as graph algorithms and machine learning. Spark runs on Mesos, standalone, cloud and Hadoop. It's possible for Spark to access distinguished and variety of data sources which includes HBase, HDFS, S3, and Cassandra[26]. The other major components of Spark includes Spark SQL, Spark streaming, GraphX, Spark Core Engine and, MLlib. RDD[25] that is Resilient Distributed Datasets is an immutable distributed collection of objects. While each dataset in RDD is further divided into logical partitions, this may be computed on different nodes of the cluster. RDDs can contain any type of Scala, Python or Java objects, including user defined classes. If evaluated all together RDD is partitioned collection of records with a read only access specification. One of the ways to create RDDS is through deterministic operations on either on stable storage or on other RDDs[3][5]. While the collection of elements in RDD is fault-tolerant, they can be operated parallel all together.

The cluster manager component handles all resources that are available on the cluster. Some of the examples of such resources are CPU and memory. The flow goes in such a way that initially the driver program starts its execution, and further it connects to a cluster manager that is actually being supported by Spark and later on it requests available executors on the working nodes. Once these resources are allocated, the application is thereby shipped to the executors eventually performing the actual computations.

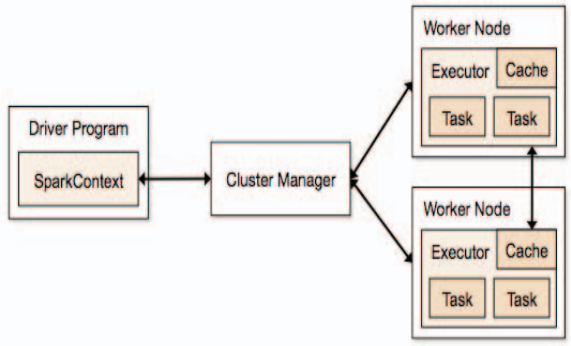

Fig.1 Execution of application on Cluster.

D. Time Series Analysis

Ross defines the statistics as "the art of learning from data"[4] in the year 2012. This means that we can learn from data for optimal decision by statistics. The main process of learning from data is to analyze data by statistical methods such as regression or time series modeling.

Fig.2 Divided Data Sets for Big Data Analysis

In this we break the complete set of big data closed to population into some sub data sets with small size in

such a way that it is closed to sample. These sub data sets are moreover proper to statistical analysis. This paper mainly focuses on Time series analysis as a statistical method for performing big data analysis. A sequence of data points that is measured normally at successive points while considering the time spaced at uniform time intervals is known as time series. Time series analysis in general consists of various methods for analyzing time series data with the objective to extract different characteristics and meaningful statistics of the data. When it comes to statistics, the main aim of time series analysis is forecasting. A Time Series[13] is a set of observations O_t, while each observation is being recorded at a specific time, consider as t. Discrete -time time series is defined as the set T of times. In discrete-time series, observations are made is a discrete set in itself. When observations are recorded continuously over some time interval, it is known as Continuous-time time series. Visualization based plotting in the form of line charts are done in order to express a time series. Time series forecasting[14] can be defined as a model designed in order to make the future predictions based on the previously observed values. Univariate or multivariate is the term used to specify the resulting time series. Many at times it has been observed that analyst prefer to use time sequence as a reference to a time series, although it is also noted that some authors refer to time sequence only when the corresponding values are non-numerical. Some of the common operations performed by time series data mining method are: clustering, novelty detection, rule discovery, motif discovery, classification and indexing.

1) Indexing: While the query time series is given indexing is done to find the most similar time series existing in a database.

2) Clustering: When the given condition consists of similar time series belonging to the same group while in contrast to this the time series belonging to different groups are also different from each, in such case clustering can be used to find the time series already existing n the database that satisfies the mentioned condition too.

3) Classification: Classification includes the task of assigning a given time series to the already existing group in such a way that it satisfies the condition of the other time series present in the group to be similar to it than the time series present in other groups.

4) Novelty Detection: When the actual expectation includes the sections of predefined base model then novelty detection is used to detect and find all the remaining sections of time series that behaves differently than the expected one.

5) Motif Discovery: It is used to detect the already unknown repeated patterns in an existing time series database.

6) *Rule Discovery:* After making the observations from one or more time series it is possible to conclude the future behavior of the time series at some specific time interval.

III. System Model

The system model has following terminals:

A. Web Interface

It consists of web application and web services for accessing analytical services. The analytical service is provided as time series analysis. The user interface for the clients is web browser with the set of tools for uploading data from the local computer or from the database, while the data can be selected from the HDFS storage and analysis of the data.

B. R Paralleled Library

Open source Apache Hadoop and Open source high level statistical programming language R[12][21] is collaborated to create a parallelized library of R analytics. Data is read using HDFS and is analysis performed using Apache Spark. The Hadoop distributed file system facilitates the user to store, read, and write in Hadoop. Since R is incompetent in the statistical analysis of large scale information, it is linked with Hadoop to create a more efficient system. One that will prove useful for several industries including stock market, artificial intelligence designing, scientific research and many more where business analytics needs to be performed. These are some of the libraries to perform time series analysis using spark and Hadoop in Distributed environment. We can achieve high performance analysis on time series data which is faster than analysis using MapReduce paradigm. We are writing parallelized library using R programming language. RSpark[16] is connector to connect R Programming Language with Spark framework and run analysis in distributed cluster nodes to achieve high performance system in general.

C. Spark SQL

In order to work with structured data Spark's package consists of Spark SQL[9][11]. Spark SQL enables querying data either using SQL or Apache Hive variant of AQL also known as HQL that is Hive Query Language[19]. It also supports various different sources of data. This includes Parquet, JSON and Hive Tables. Apart from simply providing a normal SQL interface to Spark, Spark SQL also allows developers to merge different SQL queries with the programmatic data manipulations that are supported by RDDs in Scala, Python and Java. While this all comes under a single application there for it combines SQL with complex analytics. This feature of tightly integrating the context with the rich and advance computing environment provided by Spark makes it better than any other existing open source data warehouse tool. Spark SQL was included as version 1.0 in Spark.

Before the Spark SQL came into use, Shark was an older built version of SQL-on-Spark project that modified Apache Hive to run on Spark. The older one has now been replaced by Spark SQL in order to provide better integration with different language APIs as well as the Spark engine.

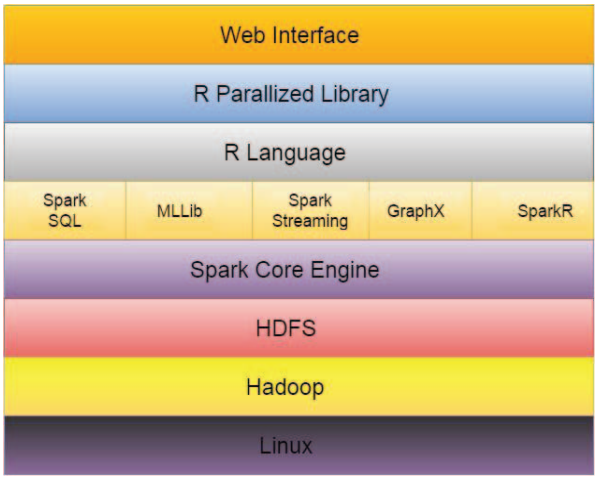

Fig.3 System Model.

D. MLlib

MLlib[6][11] is a common machine learning (ML) which is included in Spark library. It provides many types of machine learning languages like regression, collaborative filtering, classification and clustering. It also supports multiple functionalities such as data import and model evaluation. Other than this it provides few lower level machine learning primitives. This includes as a generic gradient descent optimization algorithm.

E. Spark Streaming

Spark Streaming[6][11][25] is one of the Spark components that permit the processing of live streams of data. Some of the examples of data streams include queues of messages which basically contain different status updates posted by users of a web service or log files generated by production web servers. Spark Streaming offers an API for manipulating data streams that almost is similar to the Spark Core's RDD API, hence making it easy for programmers to learn the project and enables them to move between applications that manipulate and manage data stored in memory, either on disk, or the one arriving in real time. Besides this, Spark Streaming was designed and bought forward to give the same degree of throughput, scalability and fault tolerance as Spark Core.

F. GraphX

GraphX [6][11] is a library for performing graph-parallel computations and to manipulate graphs such as a social network's friend graph. Similar to Spark SQL and Spark Streaming, GraphX extends the Spark RDD API, thereby allowing us to create a directed graph with arbitrary properties such that it is attached to each vertex and edge. Also GraphX supports a library of common graph algorithms such as Page Rank and triangle counting and various operators for manipulating graphs such as subgraph and map Vertices.

G. Spark Core

The basic functionality of Spark is considered in Spark Core. This includes different components for memory management, interacting with storage system, fault recovery, task scheduling and many more. It also consists of the API that includes Spark's main programming abstraction defined by RDDs that is Resilient Distributed Datasets. Spark Core[18][20][25] provides and enables the way to build and manipulate the collection of items distributes over many computing nodes that are represented by this RDD's.

H. HDFS

The module that helps out with the data storage spread across several storage systems is HDFS[6][23]. Also it concerns about the different information submitted basically to commodity machine in a classified fashion. One of the major functioning of HDFS module is to maintain the bandwidth of a cluster high.

I. Hadoop

The main task of Hadoop[15][23] is to store the huge volume of data across many different systems existing in the cluster and also to come up with the solution through distributed and highly scalable batch processing system. In this fashion, a voluminous workload is eventually distributed across the cluster of number of existing datasets, thereby by performing the big data analysis within a short span.

CONCLUSION

High Performance analytics of Big Data is the efficient framework for the analysis and pattern generation of processed data when compared with traditional database systems. Emerging technology like Cloud Computing can be used for storing and retrieving data as and when required which would help in memory conservation and faster execution of the system. High Performance analytics of Big Data is way better when we consider the issues like security, availability, efficiency and scalability. Beside Time Series Analysis, other additional functionalities like regression, k-means clustering, co-relation can make it one of the best analytics tool to rely upon.

REFERENCES

[1] Mohammad Nurul Azam, "R Statistical Software for Data Analysts: Past Present and Future", Global Journal of Quantitative Science, Vol. 2. No.2. June 2015.

[2] Mrigank Mridul, Akashdeep Khajuria, Snehasish Dutta, Kumar N Prasad.M.R, "Analysis of Bidgata using Apache Hadoop and Map Reduce" International Journal of Advanced Research in Computer Science and Software Engineering ,Volume 4, Issue 5, May 2014.

[3] Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma,Murphy McCauley, Michael J. Franklin, Scott Shenker, Ion Stoica, "Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing", NSDI'12 USENIX Symposium on networked design and implementation with ACM SIGOCOMM and ACM SIGOPS, SAN-JOSE,CA, April 25-27, 2012.

[4] Sunghae Jun, Seung-Joo LeeandJea-Bok Ryu, "A Divided Regression Analysis for Big Data" International Journal of Software Engineering and Its Applications Vol. 9, No. 5, 2015.

[5] Xiaoyi Lu, Md. Wasi-ur-Rahman, Nusrat Islam, Dipti Shankar and Dhabaleswar K. (DK) Panda, "Accelerating Spark with RDMA for Big Data Processing:Early Experiences", IEEE 22nd Annual Symposium on High-Performance Interconnects, 2014.

[6] Abdul Ghaffar Shoro & Tariq Rahim Soomro, "Big Data Analysis: Ap Spark Perspective", Global Journal of Computer Science and Technology: C Software & Data Engineering Volume 15 Issue 1 Version 1.0 Year 2015.
[7] Ouzor, "Interactive visualizations with R - a minireview" http://ouzor.github.io/blog/2014/11/21/interactive-visualizations.html, November 21, 2014.
[8] Vijayalakshmi Bhupathirajul, Ravi Prasad Ravud, "The Dawn of Big Data - Hbase", IT in Business, Industry and Government (CSIBIG), IEEE, ISBN 978-1-4799-3063-0, 2014.
[9] Michael Armbrusty, Reynold S. Xiny, Cheng Liany, Yin Huaiy, Davies Liuy, Joseph K. Bradleyy,Xiangrui Mengy, Tomer Kaftanz, Michael J. Franklinyz, Ali Ghodsiy, Matei Zahariay, "Spark SQL: Relational Data Processing in Spark", AMPLab, UC Berkeley, 2015.
[10] Ruizhu Huang, Weijia Xu, "Performance Evaluation of Enabling Logistic Regression for Big Data with R" IEEE International Conference on Big Data (Big Data), 2015
[11] Zhijie Han, Yujie Zhang, "Spark:A Big Data Processing Platform Based On Memory Computing", IEEE 2015 Seventh International Symposium on Parallel Architectures, Algorithms and Programming,12-14 Dec. 2015.
[12] Yanish M. Pradhananga, Shridevi C. Karande. "Blahval: Cloud Based Big Data Analytics", Proceedings of Third Post Graduate Conference on"Computer Engineering", Vol 3,ISBN: 789351072928, cPGCON 2014.
[13] Shanta Rangaswamy, Shobha G., Samir Sheriff, Satvik Neelakant, Vaishakh B N, Time Series Data Mining Tool, International Journal of Research in Computer and Communication Technology, Vol 2, Issue 10, October- 2013.
[14] Yasushi Sakurai, Yasuko Matsubara, Christos Faloutsos, "Mining and Forecasting of Big Time-series Data", ACM New York, NY,USA 2015.
[15] B.Thirumala Rao, N.V.Sridevi, V.Krishna Reddy, L.S.S.Reddy, "Performance Issues of Heterogeneous Hadoop Clusters in Cloud Computing", Global Journal of Computer Science and Technology Volume 11 Issue 8 Version 1.0 May 2011.
[16] Accessed on 3rd March 2016) SparkR R frontend for Spark [Online] Available: https://amplab-extras.github.io/SparkR-pkg/
[17] (Accessed on 8th March 2016) Spark Lightning fast cluster computing [Online] Available: https://www.packtpub.com/big-data-and-business-intelligence/mastering-apache-spark
[18] (Accessed on 8th March 2016) Spark Lightning fast cluster computing [Online] Available: https://en.wikipedia.org/wiki/Apache_Spark#Spark_Core
[19] (Accessed on 10th March 2016) Hive Querry Language [Online] Available: https://docs.treasuredata.com/articles/hive.
[20] (Accessed on 7th March 2016) Spark [Online] Available: http://spark.apache.org/
[21] (Accessed on 7th March 2016) RSpark [Online] Available: http://www.rspark.com/
[22] Josep Adler, "R IN A NUTSHELL", second edition, O'Reilly, 2012.
[23] Tom White, "Hadoop: The Definitive Guide, 3rd Edition", O'Reilly, 2012.
[24] Mike Framption, "Mastering Apache Spark", Packt, September 2015.
[25] Sandy Ryza, Uri Laserson, Sean Owen, Josh Wills, "Advanced Analytics with Spark", April 2015.
[26] Eben Hewitt, "Cassandra The Definitive Guide", O'Reilly, 2011.