International Journal of Engineering Science Invention Research & Development; Vol. III, Issue X, April 2017
www.ijesird.com, e-ISSN: 2349-6185

NEW TECHNOLOGY TO ADMINISTER HUGE VOLUME OF DATA: TRENDSPARK

Mrs. Kanchan A. Khedikar[#], Miss. Behshid I. Khairdi[*], Miss. Varsha V. Vanga[#]
[#]*Assistant Professor, Computer Science & Engineering Department, Walchand Institute of Technology, Solapur, India.*
Student Computer Science & Engineering dept., Walchand Institute of Technology, Solapur, India
[1]kanchan.khedikar@gmail.com, [2]behshidkhairdi@gmail.com, [3]varshavanga@gmail.com

***Abstract*—Big Data, in today's world is transforming our lives. But it is also placing an unprecedented burden on our computed infrastructure. As data expansion rates are increasing, effective storing, processing, and serving the growing volumes of data is becoming the need of the hour. Initially, Hadoop technology was used for handling this huge amount of data. But, now as the data is expanding exponentially, we require some new technologies. Apache Spark is one of the latest technologies that can be used to administer such huge volumes of data. This paper presents the proposed idea of implementing the application of Apache Spark named TrendSpark. TrendSpark is an application which is responsible for analyzing the big data on the social networking sites such as twitter, facebook, instagram, etc. In this paper, we mainly focus on the trending topics, news feeds, discussed on twitter for analyzing using Trendspark. Tweets are collected from the Twitter API, examined and trained to form clusters by using K means clustering algorithm. Thus, the count in each cluster can be used to find the trending topic. This paper explain the system architecture of proposed idea, various technologies used for this model and Methodology used in project.**

Keywords*—Big Data, Hadoop, MapReduce, Apache Spark, TrendSpark.

INTRODUCTION

In today's world, the data has not only become the core lifeblood of any organization but also it is growing rapidly. Such larger volumes of data are big data. Big data is often characterized by 3Vs' the extreme *volume* of data, the wide *variety* of data types and the *velocity* of data processing. The term 'big data' often refers to the use of predictive analytics or certain other advanced data analytics methods that extract value from data. Relational database management systems and desktop statistics and visualization packages often have difficulty handling big data. Standard relational databases could not easily handle big data. The dataset volume exceeds several terabytes and petabytes and if this data were stored in a relational database (RDBMS) table, it would have billions of rows.

Initially, Hadoop was one of the first popular open source big data technologies. Hadoop is an Apache project, all the components are available via the Apache open source license. Hadoop is written in Java. It doesn't maintain indexes or relationships. It breaks data into manageable chunks, replicates them, and distributes multiple copies across all the nodes in a cluster so we can process the data quickly and reliably [8]. Hadoop was used for analysis of the big data. MapReduce is a distributed compute engine provided by Hadoop. But, due to advancement more and more technologies are emerging. One such new technology is Spark. It offers more advantages over Hadoop MapReduce. It is easy to use, fast, general purpose, scalable and fault tolerant. It offers richer API and simpler programming model than MapReduce.

Spark allows an application to cache data in memory this helps in minimizing disk I/O. Spark can process terabytes of data on a cluster that may have only 100 GB total cluster memory. Spark enables to write concise code. As we know that Apache Spark is basically written in Scala, also, Scala is a JVM (Java Virtual Machine) based language. It is ideal for big data applications where speed is very important. The reason iterative algorithm run fast on Spark is its in-memory computing capability. Spark supports variety of data sources. Any data source that works with Hadoop can be used with Spark. Thus, Spark does not require you to move or copy data from these sources to another storage systems. Hence this makes it easy to switch from Hadoop MapReduce to Spark.

The upcoming sections in the paper present the literature review, architecture of the system, technologies used in the project and the flow of the data across the application. Thus, this module is basically used for speedy and efficient analysis of the big data using latest technology of Spark and giving the trending results over graphs or a website.

LITERATURE REVIEW

Big Data Analytics in the Cloud: Spark on Hadoop vs MPI/OpenMP on DIBRIS [1] by Beowulf. Jorge L. Reyes-Ortiz, Luca Oneto, and Davide Anguita. In this work, authores explore and compare two distributed computing frameworks implemented on commodity cluster architectures: MPI/OpenMP on DIBRIS that is high-performance oriented and exploits multi-machine/multicore infrastructures, and Apache Spark on Hadoop which targets iterative algorithms through in-memory computing. Researctures uses the Google Cloud Platform service to create virtual machine clusters, run the frameworks, and evaluate two supervised machine learning algorithms: KNN and Pegasos SVM. However, Spark shows better data management infrastructure and the possibility of dealing with other aspects such as node failure and data replication.

SMART GRID Technologies Apache Spark a Big Data Analytics Platform for Smart Grid [2] by Shyam R, Bharathi Ganesh HB, Sachin Kumar S, Prabaharan Poornachandranb, Soman K P. In this paper authors presents Apache spark technology. Smart grid is a complete automation system, where large pool of sensors is embedded in the existing power grids system for controlling and monitoring it by utilizing modern information technologies. The data collected from these sensors are huge and have all the characteristics to be called as Big Data. The Smart-grid can be made more intelligent by processing and deriving new information from these data in real time. This paper presents Apache spark as a unified cluster computing platform which is suitable for storing and performing Big Data analytics on smart grid data for applications like automatic demand response and real time pricing.

Mobile Big Data Analytics Using Deep Learning and Apache Spark [4] is proposed by Mohammad Abu Alsheikh, Dusit Niyato, Shaowei Lin, Hwee-Pink Tan, and Zhu Han. In this article authors presents an overview and brief tutorial of deep learning in MBD analytics and discusses a scalable learning framework over Apache Spark. Specifically, a distributed deep learning is executed as an iterative MapReduce computing on many Spark workers. Each Spark worker learns a partial deep model on a partition of the overall MBD, and a master deep model is then built by averaging the parameters of all partial models. This Spark-based framework speeds up the learning of deep models consisting of many hidden layers and millions of parameters. In this paper they used a context-aware activity recognition application with a real-world dataset containing millions of samples to validate their framework and assess its speedup effectiveness.

The Hadoop Distributed File System [3],[7],[9] Konstantin Shvachko, Hairong Kuang, Sanjay Radia, Robert Chansler. The Hadoop Distributed File System (HDFS) is designed to store very large data sets reliably, and to stream those data sets at high bandwidth to user applications. In a large cluster, thousands of servers both host directly attached storage and execute user application tasks. By distributing storage and computation across many servers, the resource can grow with demand while remaining economical at every size. We describe the architecture of HDFS and report on experience using HDFS to manage 25 petabytes of enterprise data at Yahoo!.

A Design of High-speed Big Data Query Processing System for Social Data Analysis: Using Spark SQL [5], Kiejin Park and Limei Peng. In this paper, researtures uses Spark SQL to achieve high-speed response and analyses for social big data, distributed in-memory based Spark SQL is used to construct a high-speed query processing system for social data. In Spark SQL, social data are loaded to high-speed cluster memory instead of low-speed hard disks, and thus, can increase the query performance drastically. Moreover, they used a different data processing approach based on column-oriented Spark SQL data frame rather than existing row-oriented record unit. By doing this for informal social data, high-speed query processing is possible and through evaluating the performance of the proposed query processing system, the amount of social data that are processed in a high-speed way reaches up to Terabytes(TBs) and the query processing performance (i.e., processing time) exhibits a linear pattern along with the volume of social big data.

TECHNOLOGY USED

APACHE SPARK

Apache Spark is an open source cluster computing framework. It was developed at the University of California, Berkeley, Algorithms, Machines, and People Lab (AMP Lab) to build large-scale and low-latency data analytics applications. Although, Hadoop captures the most attention for distributed data analytics, there are alternatives that provide some interesting advantages to the typical Hadoop platform. Spark is a scalable data analytics platform that incorporates primitives for in-memory computing and therefore exercises some performance advantages over Hadoop's cluster storage approach. Spark is implemented in Scala language, which provides a unique environment for data processing. Unlike Hadoop, Spark and Scala create a tight integration, where Scala can easily manipulate distributed datasets as locally collective objects. Although Spark was created to support iterative jobs on distributed datasets, it is actually complementary to Hadoop and can run side by side over the Hadoop file system. Spark also introduces an abstraction called Resilient Distributed Datasets (RDDs). An RDD is a read-only collection of objects distributed across a set of nodes. These collections are resilient, because they can be rebuilt if a portion of the dataset is lost. An RDD is represented as a Scala object and can be created from a file as a parallelized slice (spread across nodes).

SCALA

Scala is a multi-paradigm language, in that it supports language features associated with imperative, functional, and object-oriented languages in a smooth and comfortable way. From the perspective of object-orientation, every value in Scala is an object. Similarly, from the functional perspective, every function is a value. Scala is also statically typed with a type system both expressive and safe. In addition, Scala is a virtual machine (VM) language and runs directly on the Java™ Virtual Machine (JVM) using the Java Runtime Environment version 2 through byte codes that the Scala compiler generates. This setup allows Scala to run almost everywhere the JVM runs (with the requirement of an additional Scala run time library). It also allows Scala to exploit the vast catalog of Java libraries that exist, along with your existing Java code.

Scala is extensible. The language (which actually stands for Scalable Language) was defined for simple extensions that integrate cleanly into the language.

TWITTER API

Since all of the exercises are based on Twitter's sample tweet stream, it is necessary to configure authentication with a Twitter account. To do this, you will need to setup a consumer key+secret pair and an access token+secret pair using a Twitter account.

SYSTEM ARCHITECTURE

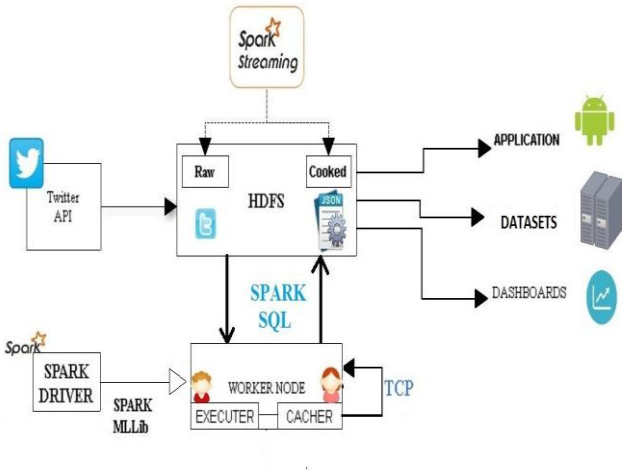

Fig. 1. System Architecture

COLLECT A DATASET OF TWEETS

Spark Streaming is an extension of the core Spark API that enables scalable, high-throughput, fault-tolerant stream processing of live data streams. In Spark Streaming, the data is ingested from Twitter API and processed using complex algorithms expressed with high-level functions. The data that is obtained is in raw format. It is the responsibility of Spark Streaming to convert this raw data into JSON format. A file of tweets is written every time interval until at least the desired number of tweets is collected. In fact, you can apply Spark's machine learning and graph processing algorithms on data streams.

EXAMINE TWEETS AND TRAIN A MODEL

Spark Streaming provides a high-level abstraction called discretized stream or DStream, which represents a continuous stream of data. Internally, a DStream is represented as a sequence of RDDs. **Discretized Stream** or **DStream** is the basic abstraction provided by Spark Streaming. Further, the Spark SQL provides SQL interface to the Spark. A Spark driver (aka an application's driver process) is a JVM process that hosts SparkContext for a Spark application. It is the master node in a Spark application. First it converts the user program into tasks and after that it schedules the tasks on the executors. Executors are worker nodes' processes in charge of running individual tasks in a given Spark job. Also, here the program examines the data found in tweets and trains a language classifier using K-Means clustering on the tweets.

APPLY THE MODEL IN REALTIME

Spark Streaming is used to filter live tweets coming in, only accepting those that are classified as the specified cluster (type) of tweets.

The core technology that is used is the Apache Spark 2.0.1. The programming languages used are Scala 2.11.8 and Java 1.8. The database used is the Spark SQL.

DATA FLOW DIAGRAM

LEVEL 0 DFD

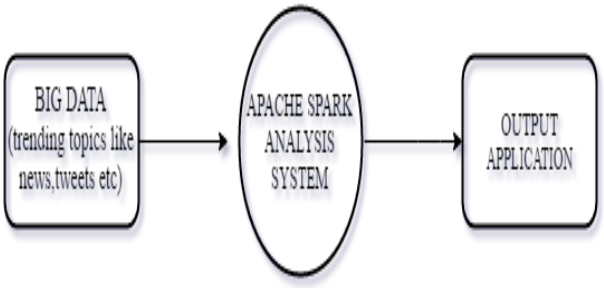

Fig. 2. Level-0 DFD

The level-0 DFD gives the basic idea, that the Big Data consisting of huge volumes of data is provided to the Apache Spark analysis system, which analyzes this data and outputs the result.

LEVEL 1 DFD

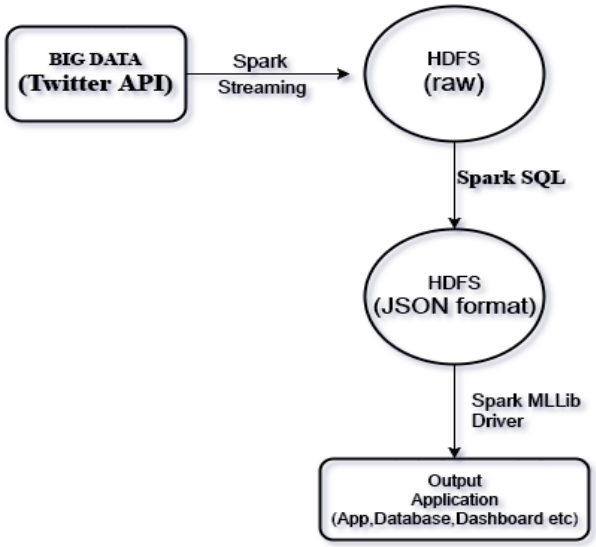

Fig. 3. Level-1 DFD

The level-1 DFD briefly explains that the tweets are obtained in the raw HDFS (Hadoop Distributed File System) from the Twitter API with the help of Spark streaming and converted to JSON (Java Script Object Notation) format by using Spark SQL. Finally, with the help of Spark MLLib (Machine Learning Library) Driver the result is output over the application.

LEVEL 2 DFD

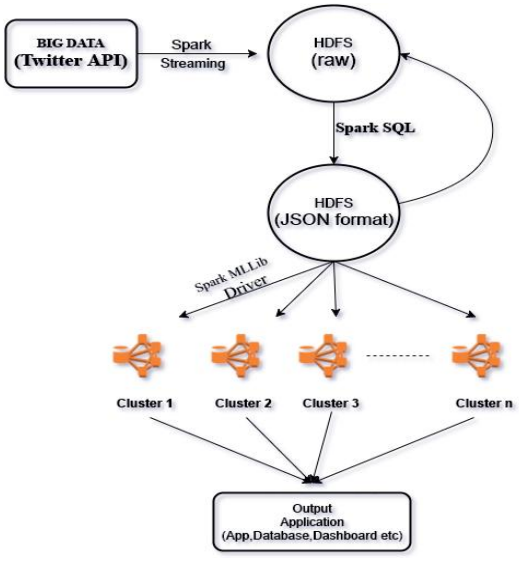

Fig. 4. Level-2 DFD

The level-2 DFD provides the detailed information regarding the flow of data across the module. The tweets are retrieved in raw HDFS

format from Twitter API by Spark streaming. The data is converted to JSON format by means of Spark SQL. The Spark MLLib Driver plays a very vital role in the forming of the clusters of data based on the algorithm. Ultimately, the trending results are output using the application.

FUTURE SCOPE

If we talk about the future scope then it depends on how innovative one could be to enhance the use of this project. But for this project, it is very useful for the future uses like It can be a one place Application for monitoring all the trends on social networks and news API. Trend analysis for monitoring consumer analytics and user traffic can help in business.

CONCLUSION

In this paper TrendSpark is a trend analysis application for Twitter big data using Apache Spark. The tweets collected from the twitter API are trained and distributed amongst the cluster. The count in each cluster is analyzed. In this way TrendSpark can be used in finding the resulting trend on the social networking site, twitter. With the help of TrendSpark, user can keep track of the top trending topics and news feeds that are discussed by a mass volume of people on the social network.

REFERENCES

Jorge L. Reyes-Ortiz1, Luca Oneto2, and Davide Anguita1, "*Big Data Analytics in the Cloud: Spark on Hadoop vs MPI/OpenMP on, Beowulf*", DIBRIS, University of Genoa. ELSEVIER, Procedia Computer Science, Volume 53, 2015, Pages 121–130.

Shyam R, Bharathi Ganesh HB, Sachin Kumar S, Prabaharan Poornachandranb, Soman K P, "*SMART GRID Technologies Apache Spark a Big Data Analytics Platform for Smart Grid*", 2212-0173 Published by Elsevier Ltd, August 2015. Available online at www.sciencedirect.com

[1] K.A.Khedikar, K.V.Kumavat, "*Role of Cloud Computing in Big Data Analytics Using MapReduce Component of Hadoop*", International Journal of Innovations in Engineering and Technology (IJIET), ISSN: 2319 – 1058, Vol. 4 Issue 1, June 2014

[2] Mohammad Abu Alsheikh, Dusit Niyato, Shaowei Lin, Hwee-Pink Tan, and Zhu Han, "*Mobile Big Data Analytics Using Deep Learning and Apache Spark*", arXiv:1602.07031v1 [cs.DC] 23 Feb 2016

[3] Kiejin Park and Limei Peng, "*A Design of High-speed Big Data Query Processing System for Social Data Analysis: Using Spark*", International Journal of Applied Engineering Research (IJAER) ISSN : 0973-4562 Volume 11, Number 14 (2016) pp 8221-8225 , Research India Publications. http://www.ripublication.com

[4] Apache Spark, "*Apache Spark–lightning-fast cluster computing*," 2016, accessed 19-February-2016. [Online]. Available: http://spark.apache.org

[5] Konstantin Shvachko, Hairong Kuang, Sanjay Radia, Robert Chansler, "*The Hadoop Distributed File System*",Yahoo! Sunnyvale, California USA {Shv, Hairong, SRadia, Chansler}@Yahoo-Inc.com

[6] Kanchan Sharadchandra Rahate(Khedikar) and Prof. L.M.R.J. Lobo. "*Modified Classification Technique Encountering Parallelization of Genetic Algorithms*" International Journal of Latest Trends in Engineering and Technology (IJLTET), ISSN: 2278-621X, Vol. 2 Issue 4 July 2013.

[7] Kanchan Sharadchandra Rahate(Khedikar) and Prof. L.M.R.J. Lobo. "*Fitness Evaluation in Parallel Environment using MapReduce*", International Journal of Computer, Information Technology and Bioinformatics (IJCITB), Vol. 1, issue 6.

[8] Ms. Kanchan Sharadchandra Rahate (Khedikar) Prof. L.M.R.J. Lobo "*A Novel Technique for Parallelization of Genetic Algorithm using Hadoop*" International Journal of Engineering Trends and Technology (IJETT) - Volume4 Issue8- August 2013. ISSN: 2231-5381.

[9] A book on *BIG DATA ANALYTICS USING APACHE SPARK* by Mohammed Guller.

https://databricks.gitbooks.io/databricks-spark-reference-applications/content/twitter_classifier/index.html

https://spark.apache.org/docs/latest/quick-start.html