IJCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Vol.-7, Issue-5, May 2019 E-ISSN: 2347-2693

Analysis of Retail Data using Apache Spark

Himani Agnihotri[1*], Bharti Nagpal[2]

[1] Department of Computer Science and Engineering, Ambedkar Institute of Advanced Communication Technologies and Research, Guru Gobind Singh Indraprastha University, Delhi, India
[2] Department of Computer Science and Engineering, Ambedkar Institute of Advanced Communication Technologies and Research, Guru Gobind Singh Indraprastha University, Delhi, India

[*]Corresponding Author: himani.a2395@gmail.com

DOI: https://doi.org/10.26438/ijcse/v7i5.11621165 | Available online at: www.ijcseonline.org

Accepted: 17/May/2019, Published: 31/May/2019

Abstract— The use of social media sites and Internet is increasing at an alarming rate. Therefore data is generated in huge amounts fraction of a second. This huge amount of data which is characterized by volume, velocity and variety is termed as big data. There is need for a framework that can process this huge amount of data and also analyze it efficiently. Apache Spark is an open source cluster computing platform which can process and analyze data efficiently. In this paper an overview and a simple example of analysis of retail data using Apache Spark is given to demonstrate its functionality.

Keywords— Big Data, Hadoop, MapReduce, Yarn, Spark, RDD

I. INTRODUCTION

Big data is data which is beyond the storage capacity and processing power of database systems. Some of the sources which generate big data are social media sites, ecommerce, transactions, search engine, etc.

Three V's which describe big data are volume, velocity and variety. 'Volume' refers to the huge amount of data. At present around quintillion bytes of data is generated each day. It is estimated that this rate will keep on increasing in future. 'Velocity' refers the large amount of data which is generated in a very short period of time. One of the major challenges today is to make use of the data which is arriving at a very fast rate. 'Variety' refers to the different formats and sources of data. The different ways to organize the data.

Apache Spark is general purpose cluster computing distributed system. It is up to hundred times faster than Hadoop MapReduce. It can perform real time, interactive, iterative, graph, in-memory as well as batch processing of data. It is written in Scala, a functional programming language. It provides Application Programming Interfaces in Scala, Java and Python[1].

The frameworks prior to Apache Spark were limited to specific functions. For example Hadoop MapReduce is limited to batch processing, Apache Storm is limited to stream processing, and Apache Giraph is limited to graph processing. Therefore we needed a general purpose processing engine that is not just limited to one function but can perform all of the above tasks.

Apache Spark was introduced in UC Berkley in 2009. It was open sourced in 2010.It was donated to Apache in 2013. It became top level project in 2014. It also set new world record in sorting in 2014. In 2015, it was the most active project at Apache.

Figure 1 shows the components which are present in Spark ecosystem. Spark Core is the main execution engine. It is used for providing support to various applications. Spark SQL provides users the ability to submit queries in SQL on top of Spark. Spark Streaming enables users to process streams of live data. Spark MLlib provides scalable machine learning library that consists of many high quality machine learning algorithms on top of Apache Spark. GraphX is a graph computation engine.

Spark Context is needed by every spark application. It is the main entry point for every spark application. Figure 2 shows the working of spark context and its components. RDD is created using spark context. The detailed explanation of RDD will be given ahead.

Driver program contains main method. It starts the execution of a process. Cluster Manager is an external service which manages the resources on the cluster. Worker node also referred to as slave node is responsible for running applications in the cluster. It consists of executors, tasks and cache. Every job submitted by user consists of multiple tasks. A task is the smallest unit of execution. An executor is a process launched on worker node that is responsible for running the task.

An RDD is an immutable, partitioned collection of elements distributed across the cluster that can be

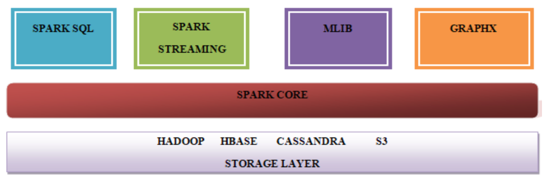


Figure 1. Spark Ecosystem Components

operated in parallel. Some of the basic operations which are available on all RDD are map, filter and persist.

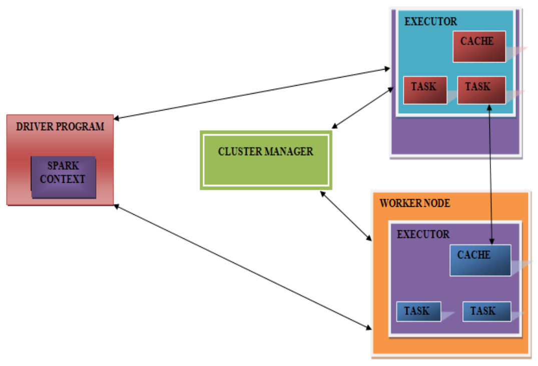


Figure 2. Spark Context

Partitions are parts of RDD which helps spark to be executed in parallel. Partitions are distributed throughout the cluster. Every task is responsible for processing one partition at a time. Partition is used to represent every input, intermediate and output data. This division of data using partitions is derived from Hadoop MapReduce.

There are two ways for creating an RDD. The first way to create RDD is by using parallelized collections. Parallelized collections are created by calling parallelize method on an existing collection in our driver program. The second way to create RDD is by using external datasets. Distributed datasets can be taken from any storage source. There are several storage sources available such as Hadoop HDFS, HBase, Cassandra, etc.

The three types of operations which can be performed on RDD are transformation, action and persistence. Transformations are set of operations which are used to define how an RDD should be transformed. Transformations are not performed immediately. They are evaluated lazily. They get executed only when an action is called. Each element of RDD is passed to a custom function and in return we get new RDD. Example map, GroupBy, filter, etc. Action returns a value to the main program or exports the value to a storage system after performing computation. Example count(), collect, Take(), etc. Persistence is used for caching the datasets in memory for future operations. Example cache(), persist(), etc.

The rest of the paper is organized as follows. Section I contains the introduction of the topics such as big data, Apache Spark, Spark Context, Spark ecosystem and Resilient Distributed Datasets; Section II contains literature review which mainly focus on the performance comparison of Hadoop Map Reduce and Spark, perform sentiment analysis of data using Hadoop and Spark; Section III describes the environmental setup in which the analysis is performed; Section IV gives the proposed approach which is used for performing analysis of Retail data; Section V provides the results and analysis of Retail data in the form of table and column chart and Section VI concludes the work with future directions.

II. Literature Review

Lei Gu and Huan Li[2] proposed a paper in which they have conducted experiments on iterative operations in order to compare the performance of Hadoop and spark using parameters like time, memory and cost. They found that Spark is generally faster than Hadoop in terms of speed but if there is no enough memory then speed advantage is of no use and it is better to use hadoop in this case.

Abdul Ghaffar Shoro & Tariq Rahim Soomro[3] proposed a paper in which they have performed analysis of real time Twitter data called as big data using Apache Spark. They have also suggested some areas which need improvement in Spark.

Abhishek Bhattacharya[4] proposed a paper which justified the characteristics of Apache Spark. They have also compared the performance of Hadoop MapReduce and Spark using sorting and showed that Apache Spark have set world record in sorting by taking lesser time than Hadoop MapReduce.

V Srinivas Jonnalagadda[5] proposed a paper in which they have given a brief introduction about Spark and also explained it’s working with Hadoop.

Priya Dahiya et al.[6] proposed a paper which shows the working and architecture of hadoop and spark, comparison between hadoop MapReduce and spark, challenges faced by map reduce and how spark works on Hadoop YARN.

Amit Palve[7] proposed a paper which uses spark framework to process live tweets from twitter API and perform sentiment analysis. In future they plan to find a method to predict the location of a tweet on the basis of tweet’s information.

Vivek Francis Pinto[8] proposed a paper in which they have given an overview of Big Data, Hadoop MapReduce, Apache Spark and also reviewed case studies in which Apache Spark has been used.

Kalyani K. Pathrikar[9] proposed a paper in which they have given a brief introduction of Apache Spark and also given use cases of Spark over Hadoop.

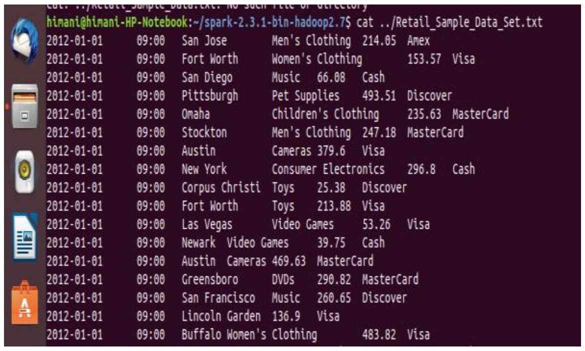
Smita M. Deshpande and R. S. Shirsath[10] proposed a recommendation system which is implemented in Hadoop as well as Spark. Using both Hadoop and Spark can process better than regular system using only Hadoop.

III. Experimental Setup

This section discusses about the experimental setup used for performing an analysis of retail data.

The operating system used in the experimental setup is Ubuntu 18.04.1. The version of Spark installed is Spark 2.3.1. The Java version used is “1.8.0_181”. The version of eclipse used is Eclipse Kepler Service Release 2.

The dataset used in the experiment is a sample retail dataset. The fields in the dataset are Date, Time, City, Product-category, Sale Value and Payment-Mode. Figure 3 shows a sample of retail dataset. Not that the fields are separated by the tab character. This is the reason that some of the fields are not aligned column-wise[11].



```
himani@himani-HP-Notebook:~/spark-2.3.1-bin-hadoop2.7$ cat ../Retail_Sample_Data_Set.txt
2012-01-01      09:00   San Jose        Men's Clothing  214.05  Amex
2012-01-01      09:00   Fort Worth      Women's Clothing        153.57  Visa
2012-01-01      09:00   San Diego       Music   66.08   Cash
2012-01-01      09:00   Pittsburgh      Pet Supplies    493.51  Discover
2012-01-01      09:00   Omaha           Children's Clothing     235.63  MasterCard
2012-01-01      09:00   Stockton        Men's Clothing  247.18  MasterCard
2012-01-01      09:00   Austin          Cameras 379.6   Visa
2012-01-01      09:00   New York        Consumer Electronics    296.8   Cash
2012-01-01      09:00   Corpus Christi  Toys    25.38   Discover
2012-01-01      09:00   Fort Worth      Toys    213.88  Visa
2012-01-01      09:00   Las Vegas       Video Games     53.26   Visa
2012-01-01      09:00   Newark  Video Games     39.75   Cash
2012-01-01      09:00   Austin  Cameras 469.63  MasterCard
2012-01-01      09:00   Greensboro      DVDs    290.82  MasterCard
2012-01-01      09:00   San Francisco   Music   260.65  Discover
2012-01-01      09:00   Lincoln Garden  136.9   Visa
2012-01-01      09:00   Buffalo Women's Clothing        483.82  Visa
```

Figure 3. Sample of Retail Dataset

IV. Proposed Approach

This section shows the steps for performing analysis on the retail data set. The steps below show the computation of the total sales by product category across all the stores. In the same way the total sales values by city can be computed. The only change is the argument i.e. city values are given instead of the product category.

1. Build the model
   a. First open eclipse and create a new Java project and package.
   b. Create a new Java class.
   c. Import necessary jar files to run the project.

2. Perform Computation
   a. First check for input argument length. The argument length should be two. The first argument should tell about the input file i.e. the retail dataset and the second file should indicate about the file in which the output will be stored.
   b. Create an object of SparkSession. The SparkSession object is necessary for starting the execution in Spark. Also give the name of the application in this step.
   c. Create a Java RDD called lines consisting of the contents given in the first argument of the input.
   d. Create a JavaPairRDD which returns a String and a Float in the output. The String is the product category and Float is the value of the sale in that category. The output is computed by applying mapToPair operation on the lines RDD and the argument of the mapToPair function is an anonymous class which overrides the call method. The call method splits the data with a tab character and returns output in the form of key value pairs. The key is the product category and value is the sale value. The return type of call method is Tuple2 which has two arguments in the form of String and Float. The created RDD is called pairData.
   e. Create a new JavaPairRDD which applies reduceByKey function on the above created pairData RDD. The argument of the reduceByKey function is an anonymous class which overrides the call method. The call method computes the sum of all the values corresponding to the key from all the nodes. The return type of the call method is a Float which corresponds to the sum value. This JavaPairRDD returns a String corresponding to the product category and a Float corresponding to its total sales values across all the stores. The created RDD is called result RDD.
   f. The final output which is stored in the result RDD is stored in the file given in the second input argument.
   g. Close the spark session by using stop() method on the SparkSession object.

3. Run the model
   a. Create a jar file of the above created java class.
   b. Open terminal and change the directory to spark.
   c. Copy jar file to Desktop.
   d. Run the command on the terminal. The command specifies the package name of the class, class name, jar file name, input and output file name.
   e. The result is a list of the product categories with the total sales value.

V. Results and Analysis

This section shows the results and the analysis obtained after performing the computation. Table 1 shows the results obtained after performing the steps given in the proposed approach. Figure 4 depicts the column chart showing the results obtained. Both the table and figure show the total sales value of five product categories.

Table 1. Total sales value of each product category

Product-Category	Sales Value
Men's Clothing	4030.89
Video Games	2573.38
Computers	2102.66
Women’s Clothing	3736.86
Sporting Goods	1952.89

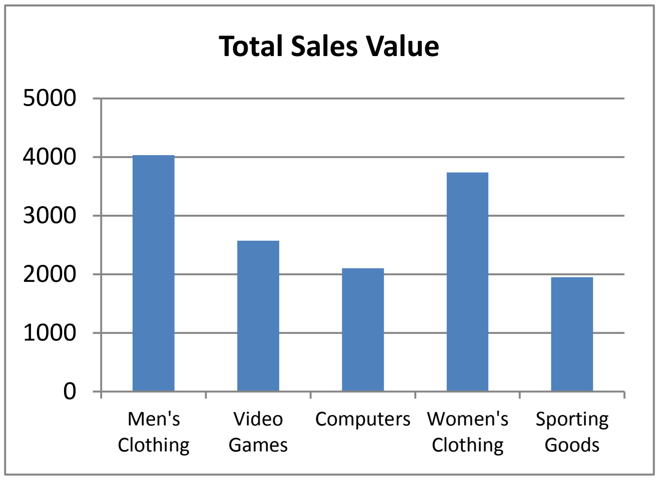


Figure 4. Column Chart showing total sales value of five product-categories

Apache Spark can be used efficiently to perform analysis of data on any parameters. The results in the form of table and chart show the total sales value of only the five product categories. The total sales value of all the product categories present in the dataset can also be calculated. In the same way the total sales values of all the cities present in the dataset can be calculated.

VI. Conclusion

In this paper a brief overview of Apache Spark is given. Also a simple example of performing analysis of Retail Data with the help of Spark is given. The analysis can be performed on various measures. This paper just shows a simple one. Apache Spark is gaining popularity day-by-day. It is better than many of the other frameworks. Therefore there is high demand to understand it thoroughly and also start to explore the applications that can make the appropriate use of Spark and all of its components. In future semantic analysis of Twitter data can be performed.

REFERENCES

[1]. Mantripatjit Kaur, Gurleen Kaur Dhaliwal, “*Performance Comparison of Map Reduce and Apache Spark on Hadoop for Big Data Analysis*”, International Journal of Computer Sciences and Engineering, Vol.**3,** Issue.**11**, pp.**66-69**, **2015**.

[2]. Lei Gu, Huan Li, “*Memory or Time: Performance Evaluation for Iterative Operation on Hadoop and Spark*”, In IEEE 10th International Conference on High Performance Computing and Communications and IEEE International Conference on Embedded and Ubiquitous Computing (IEEE 2013), pp.**721-727** , **2013**.

[3]. Abdul Ghaffar Shoro, Tariq Rahim Soomro, “*Big Data Analysis: Ap Spark Perspective*”, Global Journal of Computer Science and Technology, Vol.**15**, Issue.**1**, pp.**7-14**, **2015.**

[4]. Abhishek Bhattacharya, Shefali Bhatnagar, “*Big Data and Apache Spark: A Review”*, International Journal of Engineering Research & Science (IJOER), Vol.**2,** Issue.**5,** pp.**206-210**, **2016**.

[5]. V Srinivas Jonnalagadda, P Srikanth, Krishnamachari Thumati, Sri Hari Nallamala, ”*A Review Study of Apache Spark in Big Data Processing*”, International Journal of Computer Science Trends and Technology (IJCST), Vol. **4**, Issue.**3,** pp.**93-98, 2016**.

[6]. Priya Dahiya, Chaitra.B, Usha Kumari, “*Survey on Big Data using Apache Hadoop and Spark*”, International Journal of Computer Engineering In Research Trends, Vol. **4**, Issue.**6**, pp.**195-201**, **2017**.

[7]. Amit Palve1, Rohini D. Sonawane, Amol D. Potgantwar, “*Sentiment Analysis of Twitter Streaming Data for Recommendation using Apache Spark”*, International Journal of Scientific Research in Network Security and Communication, Vol.**5**, Issue.**3**, pp.**99-103**, **2017**.

[8]. Vivek Francis Pinto, Sampath Kini, Igneta Mcluren Dsouza, “*A Review Document on Apache Spark for Big Data Analytics with Case Studies”*, International Journal of Computer Science Trends and Technology (IJCST),Vol.**5**, Issue.**5**, pp.**99-103**, **2017**

[9]. Kalyani K. Pathrikar, Prof. Arundhati A. Dudhgaonkar, ”*Review on apache spark technology*”, International Research Journal of Engineering and Technology (IRJET), Vol.**4**, Issue.**10**, pp.**1386-1388**, **2017**.

[10]. Smita M. Deshpande, R. S. Shirsath, “*Ranking of Product on Big Data using Apache Spark*”, Sixth Post Graduate Conference for Computer Engineering (cPGCON 2017) Procedia International Journal on Emerging Trends in Technology (IJETT), **2017**.

[11]. S.N. Patil, S.M. Deshpande , Amol D. Potgantwar, “*Product Recommendation using Multiple Filtering Mechanisms on Apache Spark”,* International Journal of Scientific Research in Network Security and Communication, Vol.**5**, Issue.**3,** pp.**76-83**, **2017**.

Authors Profile

Ms. Himani Agnihotri pursued Bachelor of Technology from University of Delhi, India. She is currently pursuing Master of Technology specialization in Information Security from Guru Gobind Singh Indraprastha University, India.

Dr. Bharti Nagpal is an Assistant Professor in Ambedkar Institute of Advanced Communications Technology and Research college, Delhi. She has 18 years of teaching experience. Her main research work focuses on Information Security, Data Mining, Big Data and Web Engineering.