Big Data Analysis using R integration with Hadoop

Dr. Varun Tiwari[#], Dr. Mukta Sharma*, Dr. Vikas Rao Vadi[&]

[#]*Associate Professor, Trinity Institute of Professional Studies,*

[*]*Associate Professor & HOD, Trinity Institute of Professional Studies, Dwarka*

[&]*Professor & Director, Trinity Institute of Professional Studies, Dwarka*

[#]varuntiwari1984@gmail.com

[*]m.mukta19@gmail.com

[&]vikasraovadi@gmail.com

Abstract— **Today very large amount of data is available in the world. Therefore, there is an immediate need for maintaining, managing and accessing the large amount of data. Major IT players Like Amazon, Google has been extensively working on Cloud Computing since year 2000. In year 2006 August, Amazon introduced its Elastic Compute Cloud for Amazon Web Services. Later in year 2008, April Google released Beta version of Google App Engine. The world is familiar with cloud and in some manner using cloud, so the companies now have abundance of data. There are many storage options available for storing million user's data such as Hard Disk, CD, DVD Mobile Phone and Internet Cloud Area etc. Three types of data is available that is structured, semi- structured and unstructured. The challenge faced by the companies is of maintaining and analyzing the huge data. It is essential for the companies to retrieve the best processed information out of the data available. Now the issue arises how to analyses, process large data in cloud?**

One of the finest solutions to resolve the above-mentioned problem is R programming. The authors in this research paper have tried to focus on R programming, as it is used for analyzing and accessing the data using Hadoop. R being at a developing stage utilizes familiar scripting platforms such as python, pig for reducing processing and generate faster results. This research paper will aim at identifying the R programming Integration with Hadoop. The paper will shed light on how R programming is beneficial for analyzing Big Data using Hadoop.

Keywords- **MapReduce, R, Big Data, Cloud, Big Data Analysis, Big Data Processing, Distributed File System (HDFS), RStudio, Hadoop**

I. INTRODUCTION

R programming is an open source and it is widely used by data scientist for data analysis. R is a wonderful programming for displaying statistical data analysis represented easily in the form of graphics. R is preferred by maximum data scientists, but R has a major shortcoming as it loads all objects in main memory and as it deals with big data it is difficult to fit all data in RAM memory of a single machine. Therefore, for better results on huge data a need arises to integrate Hadoop with R programming. It is revolutionizing approach towards analysis, operation and maintenance of huge data. Information Technology companies are adopting various scripting platforms with R to reduce the time for analysing and structuring the gigantic data. R is the need of the hour for Database companies which deal with enormous data at every step. Apache Pig is an important component of Hadoop Ecosystem which reduces the coding and development time for analyzing big data. This research study furnishes information on big data and R using Hadoop various challenges related to big data. This makes available information on R architecture and specifications of various components of different layers of R Statistical processing. This study explicates the comparison of scripting platforms and provides metaphor between two platforms R and Hadoop. Existing work of big data analysis using R integration with Hadoop has also been portrayed.

II. WORKING WITH BIG DATA IN R

There are lots of packages and statistical computation providers of R language using Big Data. The paper depicts R programming using Hadoop for analysis Big Data. R is used for data processing and distributed computing, vector, list, factor and array are available to load data from any source. Same as Hadoop is a Big Data technology to handle a large amount of data. R and Hadoop both can be coordinated together for Big Data Analytic.

III. R WITH HADOOP

As already mentioned above R is one of the most chosen programing tools and especially when combined with Hadoop, R pitfalls get converted into the best tool for statistical data analysis and to convert this data analysis to interactive graph, chart and plot. In R Programming the objects are stored in the main memory on a single system. R is not scalable because only limited amount of data can be processed at a time. Therefore Hadoop is a perfect choice for loading large data instead of R. Today SAS is used with R for storing large data.

Hadoop is a distributed processing framework. It used to perform large operation and handle large data sets. It is a popular framework for Big Data using R, this is highly scalable.

IV. *BIG DATA ANALYSIS USING R INTEGRATION WITH HADOOP*

Data Scientists works on R packages and scripts for data processing. These packages and scripts need to be rewritten in java language or any such type of language. It implements Hadoop MapReduce algorithm for data processing. There are some methods used for Big data integration R with Hadoop.

1. RHadoop-: It is the best solution to process Bigdata, R with Hadoop. The user can directly take data from HBASE database systems and HDFS System. It contains five packages-:
    i. Rhbase-: It uses database management functions for HBase within R.

    ii. Rhdfs-: provides connectivity to HDFS.

    iii. Plyrmr-: provides data manipulation operations for large datasets.

    iv. Ravro-: it is used to read and write Avro files from HDFS. It is a row-oriented and data serialization framework developed for Hadoop project.

    v. Rmr2-: used for stored statistical analysis data on Hadoop.
2. Rhipe-: It is R and Hadoop integrated environment. It is providing the facility to use R library within MapReduce. It also provides data distribution and integration within Hadoop.
3. Streaming with R and Hadoop-: User directly run MapReduce using an executable script. It reads data from standard input and writes data using mapper or reducer. R scripts is fully integrated with Hadoop Streaming.
4. RHive-: This package directly used hive queries within R. It retrieves data using names, column names and table name. It provides libraries and algorithms in R to stored data in Hadoop.
5. Orch-: This is Oracle connector for Hadoop. It used to run directly MapReduce program without help of new programming language.

There are following techniques used for big data analysis with R-:

1) Sampling-: Sampling is used to decrease the size of data where data is too big.

2) Hardware-: R stored all objects in a single memory. So that the problem comes if data is very large. So, this problem solves when you increase the memory size of machine.

3) Alternative Interpreters-: There are two alternative interpreters such as pqR (pretty quick R) and Renjin which can easily run on the Java Virtual Machine.

4) Integration of High Programming Language-: R is high performing programming language. So, the small program is easily transferred from R to another language.

A. MapReduce

MapReduce is used for processing large datasets distribution on a large cluster. It allows performing massive data processing thousands of servers with Hadoop clusters. It is derived from Google MapReduce. It processes large amounts of data in parallel on large clusters of commodity hardware in a reliable, fault tolerant manner. It is divided in to two forms first is map and second is reduce. Map deal with key and reduce value pairs the data. Map and reduce run sequentially on the cluster and the output of the map becomes the input for reduce. Now explore these phases sequentially-:

a. Map: In this phase datasets are assigned to the task tracker and data functional operation will be performed over the data, assign a map key and value pairs for the output.

b. Reduce-: In this phase master node will collect all the answer, subproblems and combines them for the output.

There are five steps used for map and reduce-:

1) Preparing the Map () input: It will take input data row wise and emit key value pairs per rows. It changes explicitly according to requirements.

    Map input: list (a1, b1)

2) Execute user-provided Map () code: Map output: list (a2, b2)

3) Shuffle: it shuffles the similar keys and input them to the same reducer.

4) Execute the user given Reduce() code: In this phase developed customized reducer code will be executed, a code which was written by developer to run on a shuffled data and emit key and value.

    Reduce input: (a2, list(b2))
    Reduce output: (a3, b3)

5) Get final output: Finally, the master node collects all reducer output and combines and writes them in a text file.

B. Hive- It is Hadoop based data framework. It is developed by Facebook. It performs task using Sql Query language such as HiveQL, which are highly abstracted to Hadoop MapReduce. It is easily integrated with business intelligence and visualization tools for real-time processing.

C. R functions used in Hadoop MapReduce scripts: There are some utility functions used in Hadoop Mapper and Reducer for data processing-:

I. File- It is used to perform or create a connection to a file for read and write operation. It uses stdin and stdout function for read and write. It is used for mapping and reducing phase.

```
con <- file ("stdin", "r")
```

II. Write- It is used to write data in a file. It will be used after the key and value pair is set in the mapper.

```
write (paste (city,pagepath,sep="\t"),stdout())
```

III. print- Used to print or write data to a file.

```
print (paste (city,pagepath,sep="\t"),stdout())
```

IV. close- Closing the connection of the file after reading and writing operation is completed.

```
Close (con)
```

V. stdin: It is standard input function. It provides text mode connection. It returns the connection object.

```
conn <- file ("stdin", open="r")
```

VI. stdout- It is standard output function. It provides text mode connection. It returns the connection object.

```
Print (list(city.key, page.value),stdout())
```

VII. sink-: It drives the R output to the connection. If the connection is file or stream it will return to file or stream. Sink can also find the errors in Mapper and Reducer.

```
sink("abc.txt")\\
k <- 1:6\\
for(i in 1:k){\\
print(paste("value of k",k))\\
}sink()\\
unlink("abc.txt
")
```

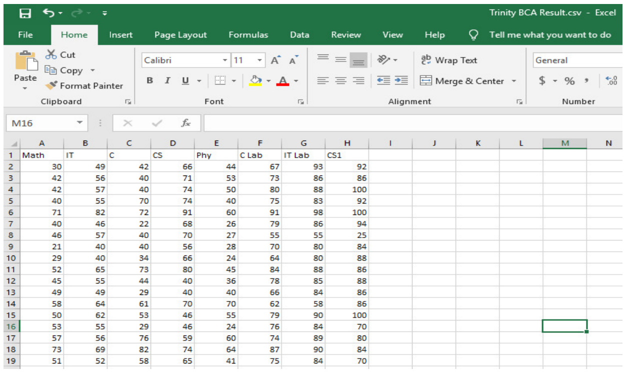
V. *Case Studies (Result Data Integration in R Studio)*

Case Study

The Education field Student Examination Result data is completely digitalized. Now this data store in PDF, excel and csv file. This excel, csv file contains Students result data. The main purpose of this case studies how to store data or read data and perform basic operation to CSV file in R Studio. After uploading or reading this file, how to perform some calculation such as Sum, Average, max, min operation using R and how to represent this data in graph or plot form. The problem is that many people are not aware of calculation and representation of data in R.

Problem:

Collecting Trinity Institute of Professional Studies (Affiliated to GGSIPU) BCA students result data and save as Trinity BCA Result.csv **(see figure 1).** The problem is that how to read csv file and perform operation in R.



	A	B	C	D	E	F	G	H
1	Math	IT	C	CS	Phy	C Lab	IT Lab	CS1
2	30	49	42	66	44	67	93	92
3	42	56	40	71	53	73	86	86
4	42	57	40	74	50	80	88	100
5	40	55	70	74	40	75	83	92
6	71	82	72	91	60	91	98	100
7	40	46	22	68	26	79	86	94
8	46	57	40	70	27	55	55	25
9	21	40	40	56	28	70	80	84
10	29	40	34	66	24	64	80	88
11	52	65	73	80	45	84	88	86
12	45	55	44	40	36	78	85	88
13	49	49	29	40	40	66	84	86
14	58	64	61	70	70	62	58	86
15	50	62	53	46	55	79	90	100
16	53	55	29	46	24	76	84	70
17	57	56	76	59	60	74	89	80
18	73	69	82	74	64	87	90	84
19	51	52	58	65	41	75	84	70

Figure 1: Trinity BCA Result.csv

Solution:

There are some steps follow for completing this task:
Step 1: Firstly, this Trinity BCA Result.csv file store or read.

> swastik <- read.csv ("Trinity BCA Result.csv”) press enter

This syntax used to read a csv file in R Studio.

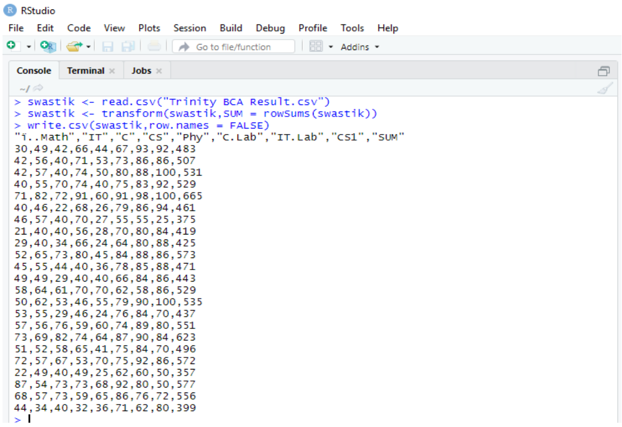
swastik <- transform (swastik, SUM = rowSums(swastik))

This Syntax is used to create SUM Field and Transform sum of rows and store in SUM Fields.

Now write these data in to Buffer.

> write.csv (swastik, row. Names = FALSE)

This is used to write data in CSV file and then show the output of the file (see figure 2).

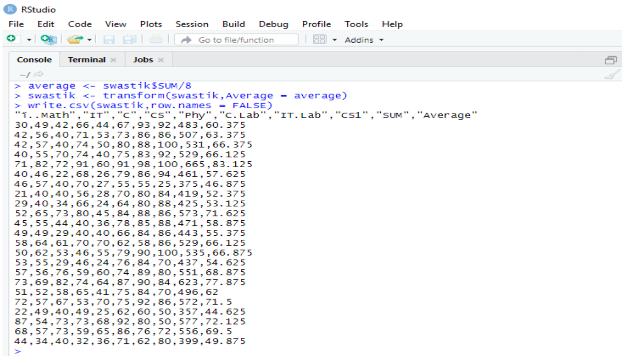


```
> swastik <- read.csv("Trinity BCA Result.csv")
> swastik <- transform(swastik,SUM = rowSums(swastik))
> write.csv(swastik,row.names = FALSE)
"ï..Math","IT","C","CS","Phy","C.Lab","IT.Lab","CS1","SUM"
30,49,42,66,44,67,93,92,483
42,56,40,71,53,73,86,86,507
42,57,40,74,50,80,88,100,531
40,55,70,74,40,75,83,92,529
71,82,72,91,60,91,98,100,665
40,46,22,68,26,79,86,94,461
46,57,40,70,27,55,55,25,375
21,40,40,56,28,70,80,84,419
29,40,34,66,24,64,80,88,425
52,65,73,80,45,84,88,86,573
45,55,44,40,36,78,85,88,471
49,49,29,40,40,66,84,86,443
58,64,61,70,70,62,58,86,529
50,62,53,46,55,79,90,100,535
53,55,29,46,24,76,84,70,437
57,56,76,59,60,74,89,80,551
73,69,82,74,64,87,90,84,623
51,52,58,65,41,75,84,70,496
72,57,67,53,70,75,92,86,572
22,49,40,49,25,62,60,50,357
87,54,73,73,68,92,80,50,577
68,57,73,59,65,86,76,72,556
44,34,40,32,36,71,62,80,399
>
```

Figure 2: Perform Sum operation in CSV

Step 2: Now find the Average of rows (see figure 3)

```
> average <- swastik$SUM/8
> swastik <- transform(swastik,Average = average)
> write.csv(swastik,row.names = FALSE)
```


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Console
Terminal
Jobs
~/

```
> average <- swastik$SUM/8
> swastik <- transform(swastik,Average = average)
> write.csv(swastik,row.names = FALSE)
"ï..Math","IT","C","CS","Phy","C.Lab","IT.Lab","CS1","SUM","Average"
30,49,42,66,44,67,93,92,483,60.375
42,56,40,71,53,73,86,86,507,63.375
42,57,40,74,50,80,88,100,531,66.375
40,55,70,74,40,75,83,92,529,66.125
71,82,72,91,60,91,98,100,665,83.125
40,46,22,68,26,79,86,94,461,57.625
46,57,40,70,27,55,55,25,375,46.875
21,40,40,56,28,70,80,84,419,52.375
29,40,34,66,24,64,80,88,425,53.125
52,65,73,80,45,84,88,86,573,71.625
45,55,44,40,36,78,85,88,471,58.875
49,49,29,40,40,66,84,86,443,55.375
58,64,61,70,70,62,58,86,529,66.125
50,62,53,46,55,79,90,100,535,66.875
53,55,29,46,24,76,84,70,437,54.625
57,56,76,59,60,74,89,80,551,68.875
73,69,82,74,64,87,90,84,623,77.875
51,52,58,65,41,75,84,70,496,62
72,57,67,53,70,75,92,86,572,71.5
22,49,40,49,25,62,60,50,357,44.625
87,54,73,73,68,92,80,50,577,72.125
68,57,73,59,65,86,76,72,556,69.5
44,34,40,32,36,71,62,80,399,49.875
>
```

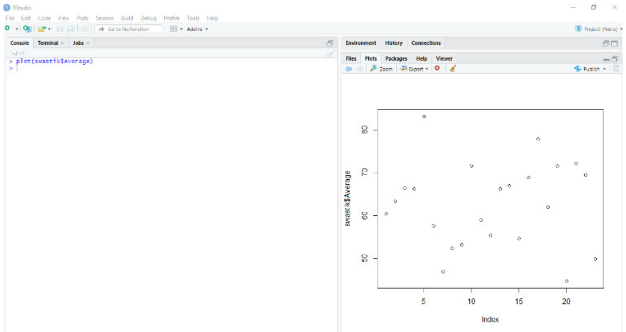
Figure 3: Perform Average operation in CSV

Step 3: Find the mean, median, min and max of marks.

```
> max(swastik$IT)
[1] 82
> min(swastik$C.Lab)
[1] 55
> median(swastik$IT.Lab)
[1] 84
> mean(swastik$CS1)
[1] 80.3913
```

Step 4: Now find the graph according to average:

```
> plot(swastik$Average)
```



```
> plot(swastik$Average)
>
```

Figure 4: Representation of Graph for average

VI.*CONCLUSION*-

R with Hadoop is a platform which provides the best data analytics solution. This can process very large data efficiently and they are illustrated graphically with beautiful visuals. It removes redundancy from the structural data. Different R packages and Scripts are there and could be used in combination with Hadoop. R-hadoop is the best approach of developing algorithm which is easily used in Hadoop. Linear or logistic regression can be shown in the form of graph and chart for better understanding of the task. Data Analysis, manage data, stored data all these tasks are easily performed by the Hadoop using R.

REFERENCES

1. Buyya, Rajkumar. 2016. "Big Data Analytics = Machine Learning + Cloud Computing." In Big Data, 7-9. Massachusetts, USA: Morgan Kaufmann Publisher.
2. Chandio, Aftab Ahmed, Nikos Tziritas, and Cheng-Zhong Xu. 2015. "Big-data processing techniques and their challenges in transport domain." ZTE Communications.
3. Cheatham, Thomas, Amr Fahmy, Dan Stefanescu, and Leslie Valiant. 1996. "Bulk Synchronous Parallel Computing — A Paradigm for Transportable Software."61-76. doi: 10.1007/978-1-4615-4123-3_4.
4. Gartner. 2016. "Gartner IT Glossary." Gartner Inc. Ghemawat, Sanjay, Howard Gobioff, and Shun-Tak Leung. 2003. "The Google file system." ACM SIGOPS Operating Systems Review 37 (5):29. doi: 10.1145/1165389.945450.
5. Ginsberg, J., M. H. Mohebbi, R. S. Patel, L. Brammer, M. S. Smolinski, and L. Brilliant. 2009."Detecting influenza epidemics using search engine query data." Nature 457 (7232):1012-4. doi: 10.1038/nature07634.
6. Hammond, Klavdiya, and Aparna S. Varde. 2013. "Cloud Based Predictive Analytics: Text Classification, Recommender Systems and Decision Support."607-612. doi: 10.1109/icdmw.2013.95.
7. Hashem, Ibrahim Abaker Targio, Ibrar Yaqoob, Nor Badrul Anuar, Salimah Mokhtar, Abdullah Gani, and Samee Ullah Khan. 2015. "The rise of –big data‖ on cloud computing: Review and open research issues." Information Systems 47:98-115. doi:10.1016/j.is.2014.07.006.
8. S.Harini, K.Jothika and K.Jayashree "A review of big data computing and cloud" in International Journal of Pure and Applied Mathematics Volume 118 No. 18 2018, 1847-1855 ISSN: 1311-8080 (printed version); ISSN: 1314-3395.
9. Anju Gahlawat ,Tata Consultancy Services Ltd "Big Data Analysis using R and Hadoop" IJCEM International Journal of Computational Engineering & Management, Vol. 17 Issue 5, September 2014 ,ISSN (Online): 2230-7893.
10. Bogdan OANCEA ,University of Bucharest , Raluca Mariana The Bucharest University of Economic Studies, "Integrating R and Hadoop for Big Data Analysis".
11. M. Wedel, R. T. Rust, and T. S. Chung. Up close and personalized: a marketing view of recommendation systems. In RecSys, pages 3–4, 2009.
12. Y. Zhang, H. Herodotou, and J. Yang. RIOT: I/O-efficient numerical computing without SQL. In CIDR, 2009.
13. N. F. Samatova. pR: Introduction to Parallel R for Statistical Computing. In CScADS Scientific Data and Analytics for Petascale Computing Workshop, pages 505–509, 2009.
14. M. Schmidberger, M. Morgan, D. Eddelbuettel, H. Yu,L. Tierney, and U. Mansmann. State of the art in parallel computing with R. Journal of Statistical Software, 31(1):1–27, June 2009.
15. M. Stonebraker, J. Becla, D. J. DeWitt, K.-T. Lim, D. Maier, O. Ratzesberger, and S. B. Zdonik. Requirements for science data bases and SciDB. In CIDR, 2009
16. A. Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka, S. Anthony, H. Liu, P. Wyckoff, and R. Murthy. Hive – a warehousing solution over a Map-Reduce framework. PVLDB, 2(2):1626–1629, 2009.

17 L. Tierney, A. J. Rossini, N. Li, and H. Sevcikova. snow: Simple network of workstations. http://cran.r-project.org/web/packages/snow/.