ISSN (Online): 2230-7893
www.IJCEM.org

Big Data Analysis using R and Hadoop

Anju Gahlawat
Tata Consultancy Services Ltd.
4 & 5 Floor, PTI Bldg, 4, Parliament St, Connaught Place, New Delhi

Abstract

The way Big data - heavy volume, highly volatile, vast variety and complex data - has entered our lives, it is becoming day by day difficult to manage and gain business advantages out of it. This paper describes as what big data is, how to process it by applying some tools and techniques so as to analyze, visualize and predict the future trend of the market. The tools and techniques described in this paper using the best of R language which is the future of the statistics and the Hadoop which is a parallel processing for the data so as to get a blend of best data model being processed over Big data parallelly. The integration of R and Hadoop give us the brand new environment where in R code can be written and deployed in Hadoop without any data movement. Using R and Hadoop helps organization to resolve the scalability, issues and solve their predictive analysis with high performance. You can have a much better deep dive over the big data when combined R and Hadoop.

Categories and Subject Descriptors
E.m - MISCELLANEOUS

General Terms: Theory, Languages

Keywords: Big Data, Data Analysis, R Language, Map Reduce, Hadoop

1. INTRODUCTION

1.1 Introduction to Big Data

Big data is a buzzword, or a catch-phrase, used to describe the massive volume of both structured and unstructured data which is difficult to process using traditional relational database and software techniques as per organization's hardware and infrastructure.

Big data is more than simply a matter of size which, according to IBM, holds three major attributes as:

Variety – Different type of data including text, audio, video, click streams, log files, and more which can be structured, semi-structure or unstructured.

Volume - Hundreds of terabytes and petabytes of information.

Velocity – Speed of data to be analyzed in real time to maximize the data's business value.

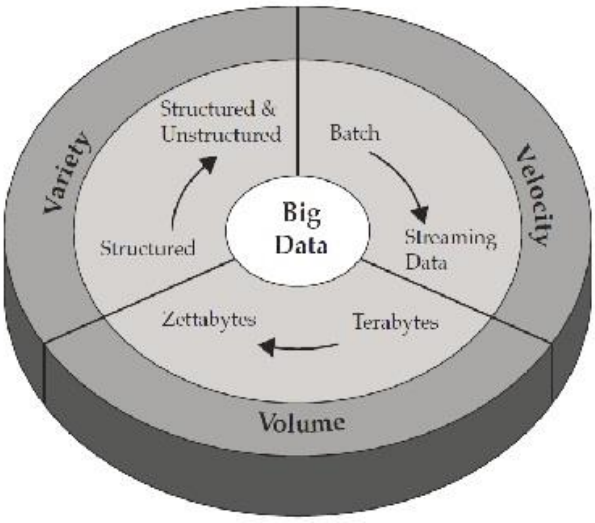

Figure 1: Big data attributes

1.1.1 Some samples of data size of few leading companies

Data Generated by NYTimes in one day

50Gb of uncompressed log files
10Gb of compressed log files
0.5Gb of processed log files
50-100M clicks
4-6M unique users
7000 unique pages with more than 100 hits
Index size 2Gb
Pre-processing & indexing time
10min on workstation (4 cores & 32Gb)
1hour on EC2 (2 cores & 16Gb)

Data Generated by Facebook in one month

30 billion pieces of content shared on facebook every month.
40% projected growth in global data generated per year vs 5% growth in global IT spending.

Let us look out **what** is big data analysis, **need** of analysis and **how** we can do analysis in optimized way through different approaches.

1.1.2 What

Big Data Analysis = Big Data + Analysis

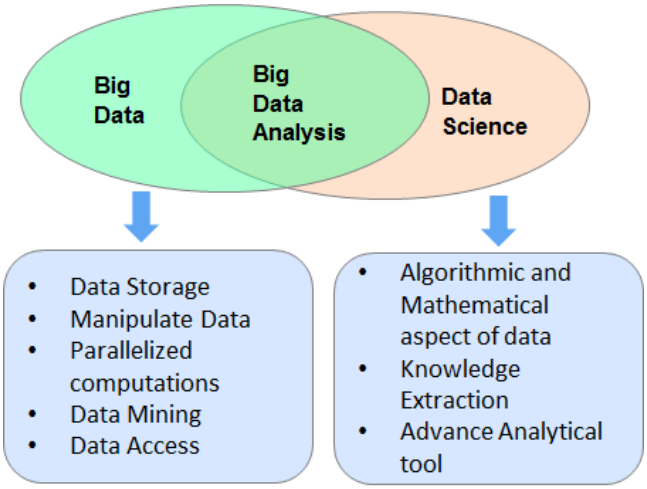

Figure 2: Big Data Analysis

We need to store, clean, brush up, apply some mathematical and algorithmic model to analyze and then beautify our [data] to get a visualized and beautiful story out of it for the senior management team to take some decisions.

1.1.3 Need

Can you imagine how your organization handles Big Data during daily operations? Just to give you an idea, consider the following scenarios:

What if a Monday morning suddenly your VP asks, “Hey, can you provide me a quick sentiment analysis for ABC news story published yesterday?"

Or, “Can you quickly draw a graph of sentiment analysis for ABC news story published yesterday for a XYZ region?”

What do you need to do so as you can respond to your VP quickly by scanning your big data?

1.1.4 How

Previously, it was the statisticians whom to play with data and come up with some models to reach out to some decision. Now it’s the data scientists whom come up with such solutions. So a data Scientist is a mixed blend of a data base expert, a statistician and a story writer. In order to make their life easier we have R language wherein we can either store data or use existing data(from some database like SQL server or oracle) and then we can perform our analysis using some predefined packages within R.

R can handle big data using ff, fbase, RODBC, RHadoop packages.

2. Big Data Analysis using R

2.1 Introduction to R

R is an open source language which is used for data modeling, manipulation, statistics, forecasting, time series analysis and visualization of data. R language uses the RAM of your machine, so bigger the RAM of your machine the bigger data you can hold for R to work upon.

We have more than 4000 different packages developed by various scholars to be used as per requirements.

The latest version of R is going to be R 3.0.2

Initially R was not used as Big Data Analysis language due to its memory limitations problems. Gradually R got some libraries like ff, ffbase, Rodbc, rmr2 and Rhdfs to handle big data . Rmr2 and rhdfs together use the power of Hadoop in order to handle big data effectively.

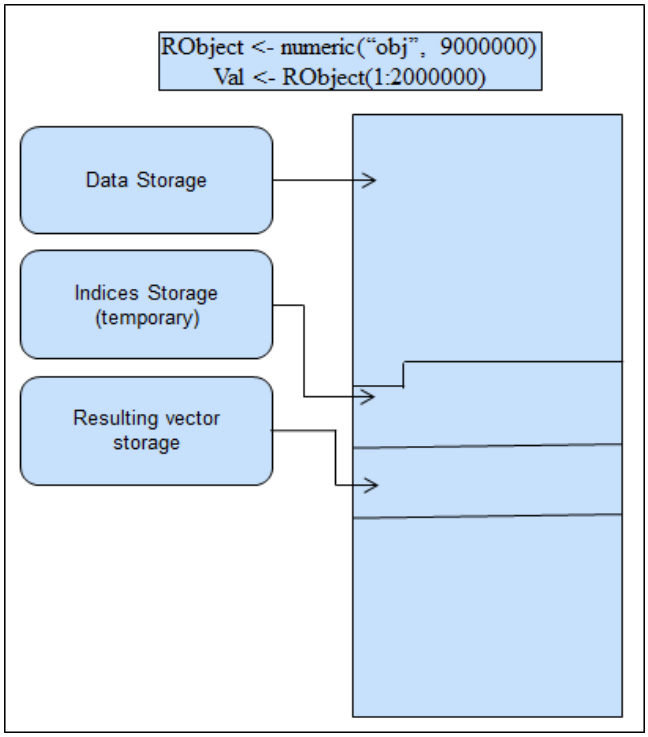

Figure 3: Data Storage with standard R object

ISSN (Online): 2230-7893
www.IJCEM.org

2.2 Big Data Analysis in R using ff, ffbase Packages

Native R stores everything into RAM. R objects can take memory upto 2-4 GB, depends on hardware configuration. Beyond this, it returns "Error: cannot allocate vector of size ……" and leaving us handicapped to work with big data using R.

Thanks to R open source, group of scholars who continuously strives in creating R packages which help us to work effectively with big data.

ff package developed by Daniel Adler, Christian Gläser, Oleg Nenadic, Jens Oehlschlägel, Walter Zucchini and maintained by Jens Oehlschlägel is designed to overcome this limitation. It uses other media like hard disk, CD and DVD to store the native binary flat files rather than is memory. It also allows you to work on very large data file simultaneously. It reads the data files into chunk and write that chunk into the external drive.

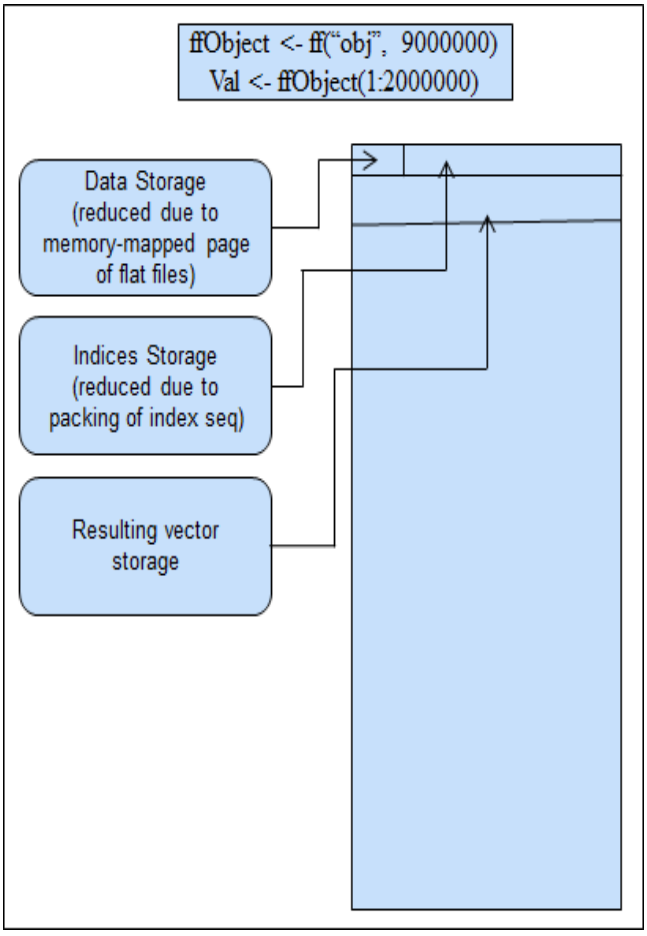

Figure 4: Data Storage with ff object

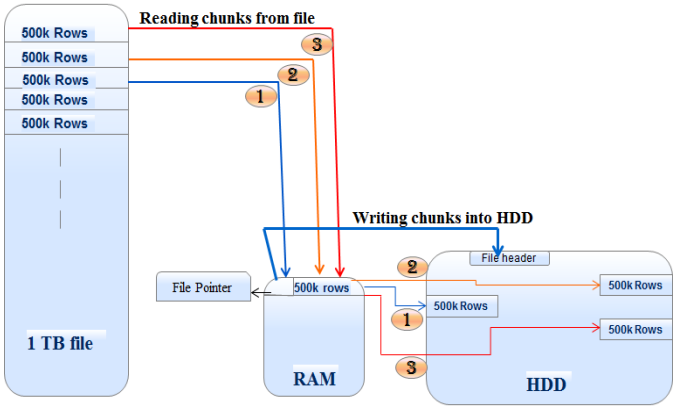

Figure 5: Functioning of ff package

```
File_chunks <- read.csv.ffdf(file="big_data.csv",
header=TRUE, sep=",", VERBOSE=TRUE,
next.rows=500000, colClasses=NA)
```

Figure 6: Read CSV file using ffdf

In the same way, we can write csv or other flat files in chunk. It reads chunk by chunk from HDD or any other external media and write it into csv or other supported format.

```
write.csv.ffdf(File_chunks, "new_file.csv")
```

Figure 7: Write CSV file using ffdf object

ff provides us the facility with ffbase package to implement all sorts of functions like joins, aggregations, slicing and dicing.

```
Merged_data =merge(ffobject1, ffobject2,
by.x=c("Col1", "Col2"), by.y=c("Col1", "Col2"), trace=T)
```

Figure 8: Join two ffdf object

```
AggregatedData = ffdfdply(ffobject,
split=as.character(ffobject$Col1), FUN= function(x)
summaryBy(Col3 + Col4 + Col5 ~ Col1, data=x,FUN=sum))
```

Figure 9: Aggregation on ffdf object

With all sorts of advantages like working with big data and less dependency on RAM, ff has few limitations, such as

1. Sometimes, we need to compromise with the speed when we are performing complex operations with huge data set.

2. Development is not easier using ff.

3. Need to care about flat files that stores in the disk otherwise your HDD or external media left with little or no space.

2.3 RODBC

Hey don't worry if you have stored your data in any relational database like SQL server. R has a privilege to connect to the SQL server (using RODBC package) and pick your data from there itself with quite an ease.

RODBC package developed by Brian Ripley, Michael Lapsley and maintained by Brian Ripley. It allow us to establish connection with different RDBMS available like SQL Server, MySQL, ORACLE, SQL lite and many more and we can fire the same SQL queries that we used to fire on respective databases. To work with RODBC, we need to create an ODBC connection as we usually do with while working with oledb.

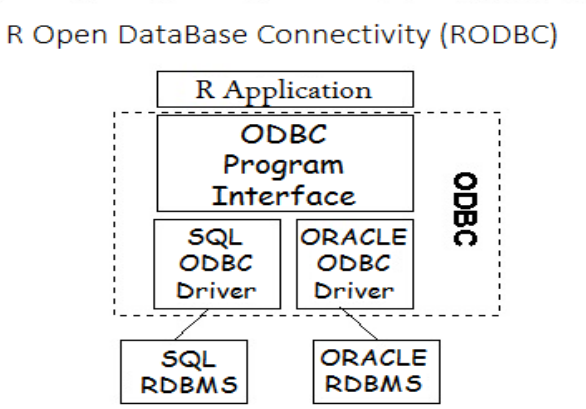

Figure 10: RODBC Connectivity

```
channel <- odbcConnect(odbc_connection, uid=userid, pwd=password);
```

Figure 11: Establish connection with Database using ODBC

```
Odbc_output <- sqlQuery(channel, "select * from table_name")
```

Figure 12: Execute SQL query in R using RODBC

ODBC act as a middle layer between database and R. R submit its request to extract data from database through ODBC which later interacts with database and finally through ODBC, data will return back to R.

It does not require HDD or external medium to store data to store data but it also has some limitations.

Again, data storage issue with RODBC package. It stores data into RAM and only limited data will be stored in memory.

2.4 Introduction to RHadoop

2.4.1 Introduction to Hadoop

Hadoop is an open source Apache software written in JAVA for running distributed applications on big data. It contains a distributed file system namely hadoop distributed file sysem (HDFS) and a parallel processing batch framework. Hadoop provides a great sense of data reliability and movement across the nodes in cluster. The core functionality lies with MapReduce which is a popular computational algorithm. MapReduce takes the approach of divide and conquer wherein the data is divided based on some mapping function and is then applied to various nodes within the cluster so as to execute parallely. Hadoop and MapReduce provides a high level of fault tolerance wherein by default data is replicated at three different data nodes (Slave nodes).

Latest stable version of Hadoop is 1.2.1another two version is going on alpha (2.0.6) and beta (2.1.0) stage.

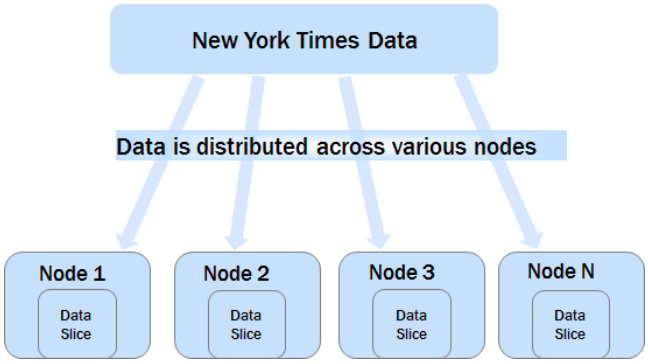

Figure 13: Example of distribute data system

Hadoop distribute its tasks and data into different nodes and each node is responsible for execution of tasks and processing of data and send result back to main node.

According to The Apache Software Foundation, the primary objective of HDFS is to store data reliably even in the presence of failures including NameNode failures, DataNode failures and network partitions. The NameNode is a single point of failure for the HDFS cluster and a DataNode stores data in the Hadoop file management system.

Yahoo has set up more than 43000 nodes hadoop cluster and Facebook has more than 100 PB(PB= 1 M GB) of data in hadoop clusters.

2.4.1.1 MapReduce – Data Reduction

MapReduce is a computational model working on divide and conquer approach using the key value pair wherein Map function divide the data set into subset and then the results are reduced to get the final output using the key.

Example of MapReduce

Below example depicts the counting and aggregation of similar kind of items belonging to a particular category in the warehouse.

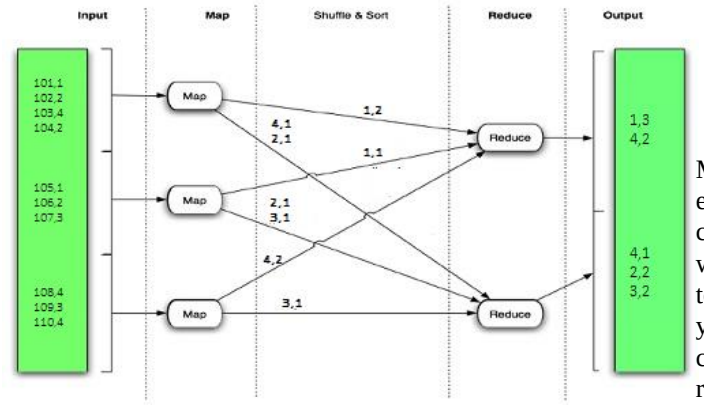

Figure 14: Example of MapReduce

Hadoop has the power for processing the data efficiently but it does not have the powerful capability for analysis. To overcome this we require Hadoop to be integrated with some statistical language like R.

2.5 Big data analysis using RHadoop

RHadoop is a collection of three R packages: rmr, rhdfs and rhbase. . rmr require Rcpp, RJSONIO, bitops, digest, functional, stringr, plyr, reshape2. Rhdfs require rJava package. We need to install these packages prior to install rmr and rhdfs respectively. rmr package provides Hadoop MapReduce functionality in R, rhdfs provides HDFS file management in R and rhbase provides HBase database management from within R. Below mentioned packages provides us the functionality of Hadoop within R

rmr2 - rmr2 provide us Hadoop MapReduce functionality in R

rhdfs - rhdfs provide us file management of the HDFS with R

rhbase - rhbase provide us database management for the HBase distributed database with R

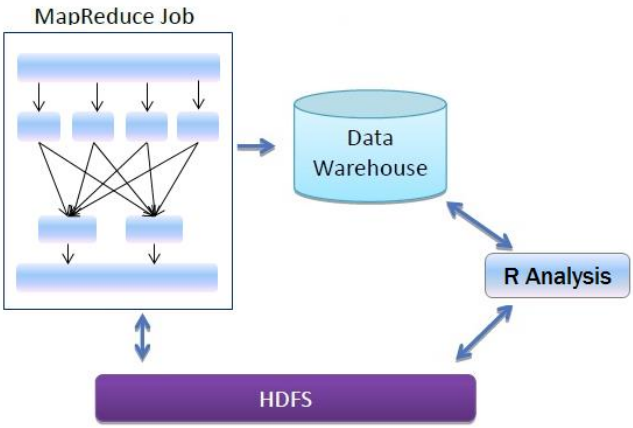

Figure 15: RHadoop Architecture

MapReduce framework is nerve system of Hadoop. As earlier mentioned, MapReduce takes the divide and conquer approach, which runs in parallel. As an analyst, we can do it on multiple dimensions. For example, we need to do a quick sentiment analysis for ABC story published yesterday. We mapped our job into smaller sets where we can applied our analytics. We process our data there and reduce our smaller data set and write our output back wither in HDFS or Hbase. On the basis of calculated result, we can draw the sentiments of users using R.

Conclusion

Rhadoop is complete set where we can process our data efficiently, perform some meaningful analysis. It removes the dependency of temp files that we used to do with ff package and any middle layer that has the limitation of memory.

Finally, there is the approach of developing algorithms that have been explicitly parallelized to run within Hadoop. For example if you wanted to do a linear or logistic regression in R on a 1TB of data stored in HDFS, this requires that

the algorithms themselves be implemented in way to use a distributed computing model. Revolution Analytics has a framework for developing these kinds of algorithms to be optimized within Hadoop.

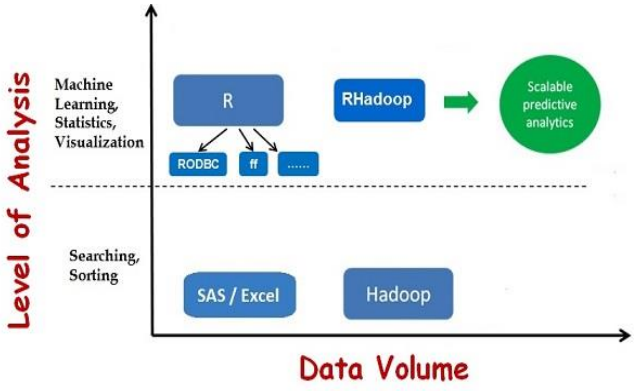

Figure 16: Technology usage according to Data volume / Level of Analysis

ACKNOWLEDGMENTS

Our thanks to IJCEM for allowing us to modify templates they had developed.

REFERENCES

http://cran.r-project.org/web/packages/ff/ff.pdf

http://cran.r-project.org/web/packages/ffbase/ffbase.pdf

http://cran.rproject.org/web/packages/RODBC/RODBC.pdf

http://rhandbook.wordpress.com/tag/rodbc/

http://cran.rproject.org/web/packages/HSAUR/vignettes/Ch_introduction_to_R.pdf

https://rhandbook.wordpress.com/tag/ffbase/