International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Memory Management in BigData

Yanish Pradhananga[1], Shridevi C. Karande[2]

[1]Student, Maharashtra Institute of Technology, Pune 411038, India

[2]Assistant Professor, Maharashtra Institute of Technology, Pune 411038, India

Abstract: ***The requirement to perform complicated statistic analysis of big data by institutions of engineering, scientific research, health care, commerce, banking and computer research is immense. However, the limitations of the widely used current desktop software like R, excel, minitab and spss gives a researcher limitation to deal with big data and big data analytic tools like IBM BigInsight, HP Vertica, SAP HANA & Pentaho come at an overpriced license. Apache Hadoop is an open source distributed computing framework that uses commodity hardware. With this project, I intend to collaborate Apache Hadoop and R software to develop an analytic platform that stores big data (using open source Apache Hadoop) and perform statistical analysis (using open source R software).Due to the limitations of vertical scaling of computer unit, data storage is handled by several machines and so analysis becomes distributed over all these machines. Apache Hadoop is what comes handy in this environment. To store massive quantities of data as required by researchers, we could use commodity hardware and perform analysis in distributed environment.***

Keywords: **M**initab, SPSS, Machine learning, IBM BigInsight, HP Vertica, SAP HANA, Pentaho, Apache Hadoop, R, Big Data.

1. Introduction

The memory management technique to deal with big data to perform linear regression and similar predictive analysis with ease and prove to be very helpful for engineering research, business, health care, scientific research, banking & finance and machine learning where complicated statistical analysis can be performed. Analysis of large data that is very complicated for traditional analytic environment is done with ease in distributed environment without undermining on the quality of the result. Entrepreneurship these days demands the gathering of information that may extend to even petabytes. Statistics based on these customer feedback data will help expand businesses and a company that has such data to its disposal, surely has a far stronger feel on the pulse of the market.

The following presentation discusses the scope of large scale data analysis and developing systems that support it for application at industrial level. Since most software packages available today focuses on the main memory of an average sized dataset in a single server, a researcher is forced to utelise vertical scalability or choose a random sample data to work upon. Vertical scalability is costly and even if this factor is overlooked, there is a limit upto which vertical scalability can be performed. But instead if one chooses a random sample, it may not always be the best representative of all the data collected.

Even complicated data analysis that require the entire data collected by the researcher, may be dealt with in this platform. These results are definitely more accurate than that given by the analysis of randomized sample [1][2]. Statistical software packages like the R, SPSS, SAS, or Matlab that are useful in the analysis of only modest quantities of data forces the user to either use more powerful computers that are rare and expensive or pick random samples that may cause results that are compromised on. Though the development of many Data Management Systems has occurred over the years, since the emphasis is on data storage, processing and simple computing, data analysis is largely left neglected. The analysis functionality is overlooked by researchers as they feel that these Data Management Systems are competent enough at data storage, data processing and simple computing. Even so, the void of an apt analytically functional statistical package is undeniable. By collaborating R and Hadoop, I intend to create a scalable platform for intense analytics and this project is work of ongoing research in generating the resultant Memory Management in Big Data.

2. Literature Survey

2.1 Open Source R Programming Language

The R[6][7][8][10] is a free software programming language used among statisticians and data miners for statistical computing, graphics and several such applications. When Ross Ihaka and Robert Gentleman created “R” in the year 1993, the appreciation and world wide acceptance [14] it generated, prompted them to make it available to the general public over the internet under the GPL. With this move R became open to a wider public and in no time, it took over much of the statistical analysis over the machine and development of the language with time made it the most eligible and updated software programming language in the industry. R was released on 29th February 2000 can be used freely, distributed and sold to any receiver who possess the same rights. The flexibility of the language due to it’s being based on a formal computer language, makes it the most sought after software programming language in the market.

R is rich in various statistical analysis packages. There are 5922 packages available in CRAN packages repository [16]. This is again increasing exponentially. There are many R users and many forums as well as groups from where you can be get concept mind as well as tutorial and support available.

2.2 Open Source Apache Hadoop

Apache Hadoop[3][4][5][7][11][12] readies us with an easy to use, dependable and accurate distributed computing software whose main purpose is to enable the researcher to

shift from single server to multiple connected machines each functioning as a separate unit as far as data storage and computing is concerned. By utilizing the Apache Hadoop software library that has several thousand computers over a cluster, one can be cent percent assured of the accuracy of analysis as each unit is so programmed as to be able to spot and rectify errors.

Hadoop stores massive quantities of data over several systems in the cluster and comes up with solutions through a highly scalable, distributed batch processing system. In this way, a large workload is distributed over the cluster of several inexpensive datasets and big data analysis is done over with in a very short time. The Apache Hadoop framework is composed of the following modules:

- Hadoop Common: This is the basic module that is concerned with libraries and ther such utilities that are essential for the optimal functioning of other hadoop modules.
- Hadoop Distributed File System (HDFS™): This is the module that concerns itself with data storage over several storage systems wherein information is submitted to commodity machines in a classified fashion. It is under the functionality of this module that the bandwidth of a cluster is kept at a high level.
- Hadoop YARN: Where YARN stands for Yet Another Resource Negotiator, is as the name suggests a task assigning and resource managing module performing at the cluster level. It is a generic platform required for any distributed application to run on.
- Hadoop MapReduce: A sub project of the Apache Hadoop project, this project is a yarn based system that is concerned with distributing input and output data in chunks for parallel processing.

2.3 RHadoop

RHadoop[7][10][14][15][17] is an open source project aimed at large scale data analysts to empower them to use the horizontal scalability oh Hadoop using the R language. ravro, plymr, rmr, rhdfs and rhbase, the 5 R packages enable its users to manage and analyze massive quantities of information using Hadoop.

- ravro – is the R package which permits the reading and writing of files in avro format, to R.
- Plyrmr – is a more recent R package that makes R and Hadoop perform in near perfect if not perfect harmony, in the analysis of higher lever plyr like data.
- Rmr - is a R package that came into being to allow users to write map reduce programs in R since it is more productive and far more easier.
- Rhdfs - is an R package that gives administration of HDFS files from within R. It uses Hadoop common to give access to map reduce file services
- Rhbase - is an R package that allows it's users basic connectivity to Hbase. Administrating a database for Hbase by using R is made possible by rhbase.

3. System Model

This system is composed of three vital components R statistical software to perform statistical analysis, Hadoop framework to distribute data in the cluster computing and Rhadoop that traverse over and link R to Hadoop. Using this system, one can load information from a local system to a hadoop library through the hadoop distributed file system that reads and analyses the input data. The four integral elements that work together to associate R language with hadoop are RAVRO, RMR, RHDFS, PLYRMR & RHBASE.

The hadoop distributed file system that facilitates the user to store, read, and write in hadoop distributed file system. Given the inability of R in the statistical analysis of massive information despite its excellent performance with moderately sized data, when linked with Hadoop, the resultant system is very promising for several industries including stock market, artificial intelligence designing and scientific research and many more where business analytics need to be performed.

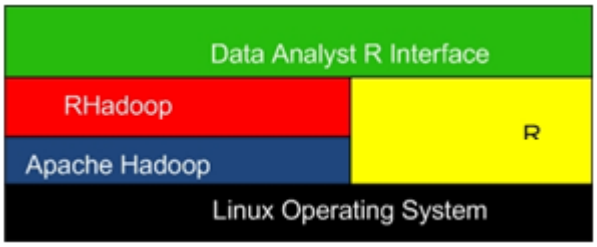

Figure 1: System Model

3.1 Memory Management

Since apache hadoop is a cluster comprising of several machines all performing on commodity hardware, looking elsewhere for data storage is out of question as it is simply possible to do horizontal scaling of memory in each worker node or even the master node.

The 2 challenges met with by R users due to its constitutional property of reading memory by default are

3.1.1 Reading data from Hadoop distributed file system

Rhdfs and Rhbase both developed as parts of the Rhadoop project helps researches using R, utilize Hadoop hdfs and Hadoop base better. Of these, Rhbase is what gives the cardinal connectivity to hadoop distributed file systems. The tables in hbase can be read, written and modified by a researcher using R.

3.1.2. Computation Memory Management

Rhdfs and by processing it either on the device or near it reduces the area of conveyance. This is done in 2 steps

- Map step: When a problem is assigned to a cluster, the master nodes distributes it to smaller questions and poses it to worker nodes that may distribute it to smaller portions and distribute it among more nodes, thereby creating a tree like pattern. When these worker nodes come up with a solution, it is transmitted back to the master node.
- Reduce step: The solution s that are passed on to the master node by the worker nodes are processed into a single solution as required by the researcher.\

As long as each mapping procedure is not intertwined with another, parallelism is a possibility not conceived yet due to the restrain in either the number of separate storage systems or the number of CPUs in close proximity to the operation site. And as long as reduction procedures share the same key and are submitted to the same reducer at the same time, a set of reducers can perform the reduction phase with ease.

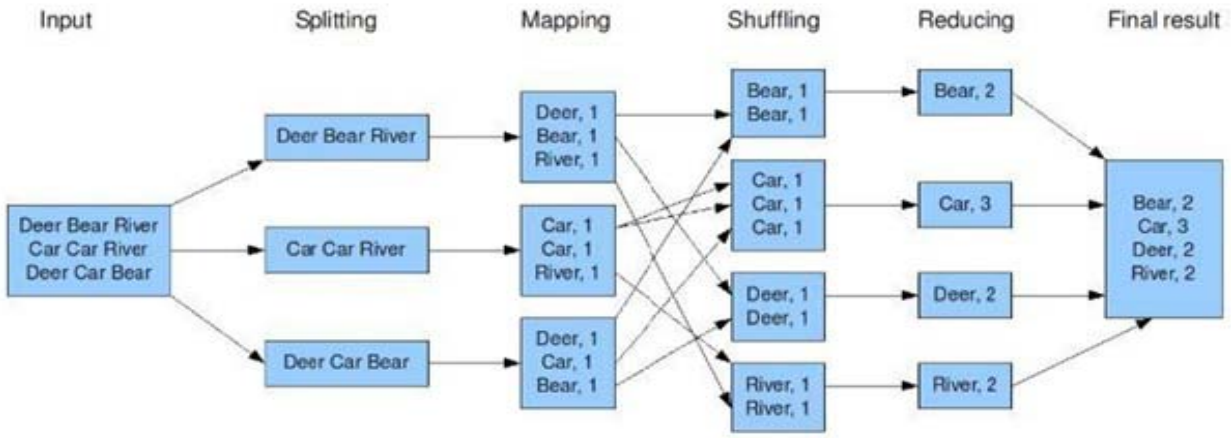

Figure 2: MapReduce task of Wordcount

A petabyte of data can be processed by a cluster using MapReduce. An advantage of the use of parallelism is that an incomplete failure of server can be resolved if the input data is still available, and even if one mapper or reducer fails, the effort can be scheduled for later.

4. Result and Discussion

The more concerning challenge has to deal with massive data as R is constitutionally designed to read the whole information in memory and scrutinize it all to reach a solution when posed with a problem, thereby creating 2 drawbacks that need be solved.

4.1 Data Memory Management

Depending on hardware configuration R objects can occupy upto 2-4 GB exceeding which, the researcher will be forced to use R and thereby analysis of big data will have to be done. To overcome these limitations, packages like ff[18], ffbase[18], bigmemory[19] etc comes to play. Ff is a package that stores large data efficiently on a disc that is quick to access. However it can map only a portion of this data n the main memory thereby acting as if in RAM and sectioning the effective virtual memory consumption per ff object. To raise performance standards to an utmost, many improvising and simplifying approaches like preprocessing and virtualization can be made use of.

Ff packages could use external storage devices like hard disk, CD, DVD etc to store binary files instead of using it's memory. It allows us to work on massive data at one go by reading and writing file in a chunk. The cost of increasing the memory of a system or cluster is very high and thus it is more reasonable to use R bearing in mind the fact that while performing analysis, the entire data in memory will be taken into consideration.

Memory management is highly efficient in R ff and ffbase. But these packages are compatible only with hadoop distributed file systems

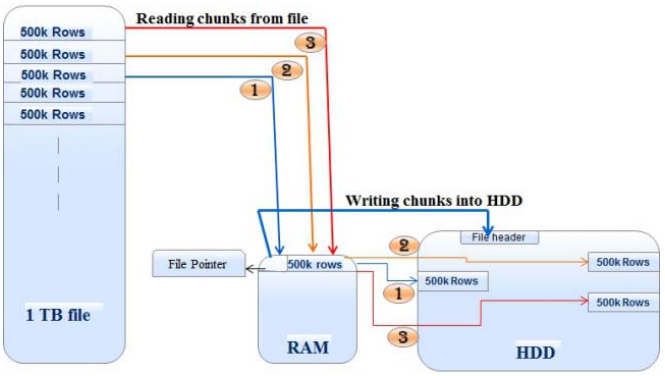

Figure 3: Working Mechanism of ff – package

4.2 Computational Memory Management

Bigmemory, Biganalytics[20] and Biglm[21] are programmes that can boost memory limitations. Biganalytics has in it multiple analytic functions. However ff is not compatible with hadoop and Biglm that provides functions for linear regression is not compatible with cluster systems.

```
library(ff)
library(ffbase)
library(biglm)
options(fftempdir = "c:/bin/")
regh<-
read.table.ffdf(file="d:/sampledata/201201hourly.txt",sep=",
",FUN="read.csv",header=T,VERBOSE=T,
next.rows=401000, colClasses=NA, na.strings="NA")
reg<-
bigglm(WBAN~Time+StationType+SkyConditionFlag+Visib
ility+DryBulbCelsiusFlag+DewPointCelsiusFlag,
data=regh)
system.time(reg<-
bigglm(WBAN~Time+StationType+SkyConditionFlag+Visib
ility+DryBulbCelsiusFlag+ DewPointCelsiusFlag,
data=regh))
object.size(regh)
```

nrow(regh)
ncol(regh)

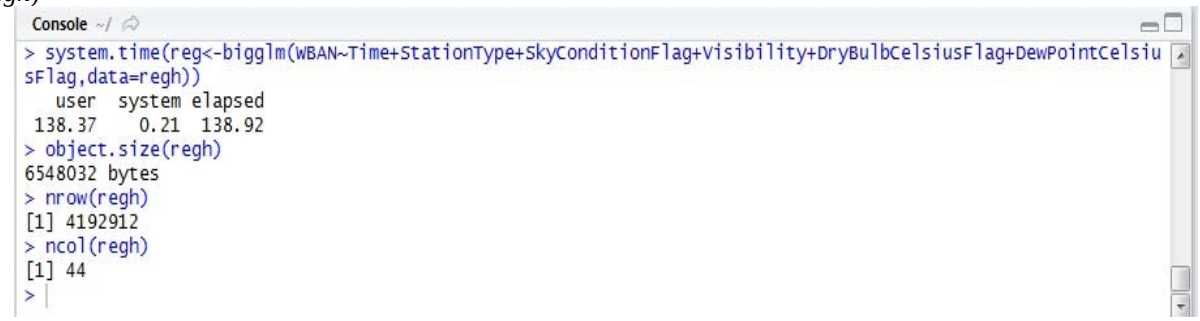

Figure 4: Showing time required to perform linear regression and showing number of row and column of file.

Above linear regression analysis is performed and result is measured in core2duo 2.0 GHz processor with 3 Gb of RAM. The text file is a weather data of one month with file size of 509 Mb and contain 4192912 rows and 44 columns. Assume if the file size is 509 Mb for one month then it'll be around 509*12Mb for 1 year. Definitely, people want to perform analysis in data of entire year and even for decade too. To perform analysis of large file is possible using these packages but it is not good practice when file size reach to terabytes and petabytes and working with single node to perform analysis where vertical scaling of memory is limited.

5. Conclusion and Future Work

With the help of data memory management and computational memory management techniques handling large scale information has taken a new turn by utterly changing memory requirement. The hurdles in the processing of large data have thus been decreased. Neither are we forced to deal with an inappropriate random sample and compromise over an incorrect statistical result, nor are we forced with the limitation of vertical scaling that has a dead end. With the addition of kmean, correlation, market basket, time series and other such functionalities, the above discussed model becomes all th more potent. By combining opensource R and opensource Apache hadoop, a cluster attains the marvel that only a crossover between a cluster framework package and a highly competent statistical programming language can achieve. The security, availability, efficiency and scalability it promises cannot be compared to traditional database systems. when additional programs like business intelligence or a link up with systems like oracle or SAP will without doubt make it the best analytic tool available in the industry. It would also prove useful if in future the cluster could maintain its optimization with a lesser number of worker odes. In today's hyper competitive world where time is money if we can turn to cloud based big data analytics when vital analysis needs to be performed in a short time, big wealth too can be sapped in a short time.

References

[1] Roger S. Barga, Jaliya Ekanayake, Wei Lu, "Project Daytona: Data Analytics as a Cloud Service", IEEE 28th International Conference on Data Engineering, 2012.
[2] Nikolay Laptev, Kai Zeng, Carlo Zaniolo., "Very Fast Estimation for Result and Accuracy for Big Data Analytics: the EARL System", IEEE's, ICDE Conference 2013.
[3] (Accessed on 12th November 2013). Hadoop [Online] Available: http://hortonworks.com
[4] (Accessed on 14th November 2013) Hadoop [Online] Available: http://en.wikipedia.org/wiki/Apache_Hadoop
[5] T (Accessed on 14th November 2013) Hadoop [Online] Available: http://hadoop.apache.org/
[6] (Accessed on 12th December 2013) R [Online] Available: http://www.revolutionanalytics.com
[7] Vignesh Prajapati, Big Data Analytics with R and Hadoop, Packt publication, 2013.
[8] (Accessed on 18th January 2014) R [Online] Available: http://cran.r-project.org/
[9] Q. Ethan McCallum & Stephen Weston Parallel R, O'Reilly, 2012.
[10] Josep Adler, R IN A NUTSHELL, second edition, O'Reilly, 2012.
[11] Paul C. Zikopoulos, Chris Eaton, Dirk Deross, Thomas Deutsch, George Lapis, Understanding Big Data Analytics for Enterprise Class Hadoop and Streaming Data, Mc Graw Hill, 2012.
[12] Tom White, Hadoop:The Definitive Guide, 3rd Edition, O'Reilly, 2012.
[13] Sudipto Das, Yannis Sismanis Kevin, S. Beyer, Rainer Gemulla, Peter J. Haas, and John McPherson, "Ricardo: Integrating R and Hadoop", In Proc. SIGMOD'10, Indianapolis, Indiana, USA, June 6–11, 2010.
[14] (Accessed on 19th November 2013) RHadoop [Online] Available: https://github.com/RevolutionAnalytics/RHadoop/wiki
[15] (Accessed on 15th December 2013) MapReduce [Online] Availabe: https://github.com/RevolutionAnalytics/rmr2/blob/master/docs/tutorial.md
[16] (Accessed on 26th October 2014) Cran Packages Repository, [Online]. Available: http://cran.r-project.org/web/packages/.
[17] Antonio Piccolboni, "Rhadoop". [Online]. Available: https://github.com/RevolutionAnalytics/RHadoop/wiki. [Accessed: Oct. 11, 2014].
[18] (Accessed on 2nd October 2014) ff ffbase [Online] Available: http://cran.r-project.org/web/packages/ff/index.html
[19] (Accessed on 2nd October 2014) bigmemory [Online] Available: http://cran.r-project.org/web/packages/bigmemory/index.html

[20] (Accessed on 2nd October 2014) biganalytics [Online] Available: http://cran.r-project.org/web/packages/biganalytics/index.html
[21] (Accessed on 3rd October 2014) biglm [Online] Available: http://cran.r-project.org/web/packages/biglm/index.html

Author Profile

Yanish M. Pradhananga received the Bachelor degree of Electronic and Communication Engineering in 2010. He is pursuing his Master Degree of computer engineering from MIT Kothrud, Pune University. His main topics of master degree are distributed computing, data mining, big data analytics, cloud computing.

Shridevi S. Karande has completed Bachelor of Engineering degree from Nanded University, India and Master of Engineering degree in Computer from University of Pune, India. She is Assistant Professor at Maharashtra Institute of Technology, Pune. Her field of interest is cloud computing, Mobile cloud Computing, Distributed System, Operating System and she has 11 years of teaching experience.