Large Scale Data Analysis Techniques

Madhavi Vaidya Dr. Shrinivas Deshpande Dr. Vilas Thakare

Abstract :- **Large scale data analysis is increasingly important in both the academics and enterprise. Statistical languages provide rich functionality and ease of use for large data analysis. Hadoop has changed the economics and the dynamics of large scale computing. It enables scalable and cost effectively. To collect the insights from this data, R is very amazing tool which allows running advanced statistical model on data. This paper gives an overview of large scale data analysis by Hadoop and using R on Hadoop.**

Keywords : **R, Hadoop, MapReduce, Google**

I. INTRODUCTION

Big data is a vague term with many different definitions. The key thing to remember is that in this day and age, big data is distributed data. This means the data is so massive it cannot be stored or processed by a single node. The days of buying a single big iron server from IBM or Sun to handle all your business intelligence needs are long gone. It’s been proven by Google, Amazon, Facebook, and others that the way to scale fast and affordably is to use commodity hardware to distribute the storage and processing of our massive data streams across several nodes, adding and removing nodes as needed.[1] This paper is segregated into introduction of R language in 1st part. 2nd part contains the use of R on Hadoop. The analysis of large data is described in 3rd part using MapReduce and R. Later the emphasis is given on the different applications of R. In the last part of the paper the analysis is done on the basis of how large data can be analyzed with R. The illustration is given about where does the R lack and how it is beneficial when R is used onto Hadoop.

A. Distributed File Systems

Distributed File systems (DFS) have been widely used by search engines to store the vast amount of data collected from the Internet because DFS provides a scalable, reliable and economical storage solution. Search engine companies also have built parallel computing platforms on top of DFS to run large-scale data analysis in parallel on data stored in DFS.

The purpose of this paper is to consider the choice to make the analysis of the large data using Hadoop or using R on Hadoop. Hadoop comprises a distributed file system called HDFS bundle with an implementation of Google’s MapReduce paradigm. [2]

The volume of data that enterprises obtain every day is increasing exponentially. It is now possible to store these vast amounts of information on low cost platforms. Such low cost platforms such as Hadoop can be used for such purpose. To collect the insights from this data, R comes into the picture. R is very amazing tool which allows running advanced statistical model on data. It translates the derived models into colorful graphs and visualizations and executes a lot of more functions related to data science. [3]

B. R Language

R is an open source software package to perform statistical analysis on data. It is a programming language used by scientist, statisticians and others who need to make statistical analysis of data and gather key insights from data using mechanism such as regression, clustering, classification and text analysis. [4]

R is registered under public domain called GNU project which is similar to commercial S language developed under Bell Labs by John Chambers and his colleagues. R can be considered as different implementation of S and is much used as an educational language and research tool. [5]

The main advantage of R is that it is a freeware and quite similar to other programming packages such as Matlab which is not a freeware, but more user-friendly than programming languages such as C++ or Fortran.

R is a wide variety of statistical, machine learning and graphical techniques , and is highly extensible. R has various built in as well as extended functions for statistical, machine leaning and visualization tasks as below -

- Data Extraction
- Data Cleaning
- Data Loading
- Data Transformation
- Statistical Analysis
- Predictive modeling
- Data Visualization

II. ANALYSIS OF LARGE DATA BY HADOOP AND R

The phenomenal growth of internet based applications and web services in last decade have brought a change in the mindset of researchers. There is an improvement in the traditional technique to store and analyze voluminous data. The organizations are ready to acquire solutions which are highly reliable. [7]MapReduce proposed by Google is a programming model and an associated implementation for large-scale data processing in distributed cluster. Using an index of the web as documents requires continuously transforming a large repository of existing documents as recent documents arrive. Storage of data processing tasks can’t be done easily by databases. Google’s indexing system stores tens of petabytes of data and processes billions of updates per day on thousands of machines. In such cases, MapReduce plays an important role.[8] Hadoop comprises a distributed file system called HDFS bundled with an implementation of Google’s MapReduce paradigm Hadoop operates directly on raw data files; HDFS takes care of the distribution and replication of the files across the nodes in the Hadoop cluster. Data processing is performed according to the MapReduce paradigm [9]

In most organizations, data is always growing, changing, and manipulated and therefore time to analyze data significantly increases. To process the large and diverse data sets, graph data structures can be processed by Hadoop and MapReduce.[10] The Map Reduce model is applied to large batch-oriented computation, which is connected primarily to job completion in proportion to time. The Map Reduce framework by Google and open-source Hadoop’s system emphasize the usage through a batch-processing implementation strategy: the whole output of each map and reduce stage is materialized to stable storage before it can be consumed by the next stage. This materialization allows for a simple that is critical in large deployment, which have a high probability of slowdowns or failures at worker nodes.[11]MapReduce proposed by Google is a programming model and an associated implementation for large-scale data processing in distributed cluster. [12]

Big Data has to deal with large and complex datasets that can be structured, semi-structured or unstructured and will typically not fit into memory to be processed. They have to be processed in place, which means that computations have to be done where the data resides for processing. They typically would mention the 3 Vs model of Big Data, which are velocity, volume and variety.

Velocity refers to the low latency, real time speed at which the analytics need to be applied. A typical example could be a continuous stream of data originating from a social networking site or aggregation of different sources of data.

Volume refers to the size of the dataset. It may be KB,MB, GB and TB or PB based on the type of the applications that generates or receives the data.

Variety refers to the various types of data that can exist, for example, text, audio, photos etc. Big Data includes datasets with sizes. [3]

Big Data Analytics describes the efficient use of a simple model applied to volumes of data that would be too large for the traditional analytical environment. As the amount of data—especially unstructured data—collected by organizations and enterprises explodes, Hadoop is emerging rapidly as one of the primary options for storing and performing operations on that data.

A. Big Data Analytics With Hadoop

The co-ordination of R and Hadoop seems a natural one. Both are open source projects and both are data driven. But there are some fundamental challenges that need to be addressed in order to co-ordinate with each other.

Iterative vs. batch processing - If we look at how most people do analytics, it is an iterative process. When the user wants to work with R for analysis of big data start with a hypothesis, explore and try to understand the data, try some different statistical techniques, drill down on various dimensions, etc. This is what makes R such a powerful tool, and an ideal environment for performing such analysis.

Hadoop on the other hand, is batch oriented where jobs are queued and then executed, and it may take minutes or hours to run these jobs.

In-memory vs. in parallel - Another fundamental challenge is that R is designed to have all of its data in memory and programs in Hadoop (map/reduce) work independently and in parallel on individual data slices. [13]

B. Large Scale Data Management Systems With Mapreduce

The MapReduce programming model was introduced in Google in 2004. This model allows programmers without any experience in parallel coding to write the highly scalable programs and hence process voluminous data sets. MapReduce framework is the processing support of Hadoop ecosystem. This framework allows the specification of an operation to be applied to a huge data set, divide the problem and data, and run it in parallel.

C. Reducing The Large Data By Mapreduce

The processing in Hadoop ecosystem is done by MapReduce Framework. It allows the specification of an operation to be applied to a huge data set, which divides the problem, and then the tasks are executed in parallel. Generally, all these tasks are written as MapReduce jobs in Java or Python. The outputs of these jobs are written back to either HDFS or HBASE. R can be used to perform the analysis of the data.

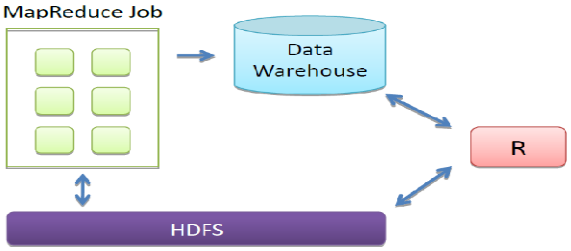

Figure 1: Data Analysis by R [13]

III. Big Data Analytics With Mapreduce And R Together

To meet the challenges of the large data analysis we need to understand data storage in Hadoop, how it can be leveraged from R, and why it is important. The basic storage mechanism in Hadoop is HDFS (Hadoop Distributed File System). For an R programmer, there is a chance to read/write files in HDFS from a standalone R Session is the first step in working within the Hadoop ecosystem. Although still bound by the memory constraints of R, this capability allows the analyst to easily work with a data subset and begin some ad hoc analysis without involving other parties. It also enables the R programmer to store models or other R objects that can then later be recalled and used in MapReduce jobs. When MapReduce jobs finish executing, then the results are written to HDFS. Executing R code in the context of a MapReduce job promotes the kinds and size of analytics that can be applied to huge datasets. Problems that fit into work in parallelized scenarios.

For a use case given as below: Scoring a dataset against a model built in R. This involves pushing the model to the Task nodes in the Hadoop cluster, running a MapReduce job that loads the model into R on a task node, scoring data either row-by row (or in aggregates), and writing the results back to HDFS. In the most simplistic case this can be done with just a Map task. This simulates the “apply” family of operators in R. Other tasks such as quintiles, crosstabs, summaries, data transformations and stochastic calculations fit well within this paradigm, Revolution Analytics. These implementations don’t make any assumptions about how the data is grouped or ordered. [13]

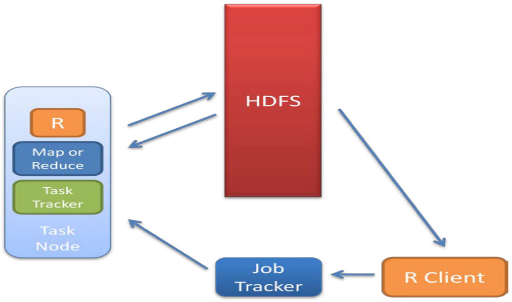

Figure 2 : Data Analysis by MapReuce and R on HDFS [13]

The strengths of R lie in its ability to analyze the data using a rich library of packages but fall short when it comes to working on very large datasets.

The limitations of R are as below –

- Requires installation of R on all Task Tracker nodes
- Does not automatically parallelize algorithms
- Different slot/memory configurations is recommended to leave memory and CPU resources for R [14]

The strength of Hadoop on the other hand is to store and process very large amounts of data in the TB and even PB range. Such vast datasets cannot be processed in the memory. The option could be to run the analysis of a data on limited chunks also known as sampling or to correspond the analytical power of R with the storage and processing power of Hadoop. Executing R code in the context of a MapReduce job elevates the kind and size of analytics that can be applied to huge datasets.

IV. Different Applications Of R

The R community is very active in improving the scalability of R and there are dozens of approaches that aim at parallelizing R across multiple processors or nodes.

High level task and data parallel computing systems for parallelizing R are usually built on top of a message passing package and are easier to use. The most popular representative R package of this type is SNOW(for Simple Network of Workstations)[15] it provides functionality to spawn a cluster, to distribute values across a cluster and to apply it in parallel a given function to a large set of alternative arguments. It can be used for task parallelism.

DISTRIBUTED R

Distributed R simplifies large-scale analysis by extending the R language. R is a single-threaded language, which limits its utility for Big Data analytics. Distributed R allows users to write programs that are executed in a distributed fashion. That is, developer-specified program components can run in multiple single-threaded R-processes. The result dramatically reduces execution times for Big Data analysis.

Distributed R is a system for large scale machine learning and graph processing. It enables and accelerates complex, big-data analysis. Starting from the open source R language and system, it adds reliable distributed processing, efficient computation over sparse datasets, and incremental processing. [6]

Many applications need to perform advanced analytics such as machine learning, graph processing, and statistical analysis on large-amounts of data. While R has many advanced analytics packages, the single-threaded nature of R limits their utility for Big Data. Distributed R extends standard R in two directions:

Distributed data structures – Distributed R stores data across servers and manages distributed computation. Users can run programs on terabyte scale datasets by simply adding more servers.

Programmers can use Distributed R to implement code that runs in parallel. Users can leverage a single multiple core computer or a cluster of computers to obtain dramatic improvement in application performance.[5]

V. Architecture Of Distributed R

Distributed R provides distributed data-structures to store in-memory data across multiple computers. These data-structures include distributed arrays (darray) and distributed data-frames (dframe).

Distributed R consists of a single master process and multiple workers. Logically, each worker resides on one server. The master controls each worker and can be co-located with a worker or on a separate server. Each worker manages multiple local R instances. The following figure shows an example cluster setup with two servers. The master process runs on server A and a worker runs on servers A and B. Each worker has three R instances.

VI. RHadoop

Focus of RHadoop is the tight integration of core Hadoop components. RHadoop is beneficial for a massively distributed systems and will work on full data sets instead of sample sets. [16] It has been stated here that the Hadoop Distributed File System can be accessed from R and written into R dataframes. The data can be read from Hadoop Distributed File System to R dataframe.

Hadoop Command Line Interface and rhdfs equivalent

- hadoop fs –ls /
- hadoop fs –mkdir /user/rhdfs/ppt
- hdfs.mkdir("/user/rhdfs/ppt")
    - hadoop fs –put 1.txt /user/rhdfs/ppt/
- localdata - system.file(file.path("unitTestData","1.txt"),package= "rhdfs")
    - hdfs.put(localData,"/user/rhdfs/ppt/1.txt")
- hadoop fs –get /user/rhdfs/ppt/1.txt 1.txt

VII. Analysis And Discussion

In this paragraph, we will discuss the use of R, in which conditions it will be appropriate. R is an open-source statistical language primarily used for data analytics. It provides a wide collection of statistical and graphical techniques and is highly extensible. R is becoming increasingly popular for sophisticated data analysis that goes beyond what can be offered by more standard business intelligence (BI) packages.[17] R is increasingly preferred as a replacement for other analytical solutions like SAS.R is R is powerful and flexible, with a rich set of statistical algorithms and graphical capabilities. There are some disadvantages of R have been found out by the author in [20][18]However, as it is single threaded and in-memory, it makes almost impossible to scale to large data sizes. Due to this limitation, data scientists usually rely on sampling the big data-sets sitting on a platform, and then performing analytics on the reduced data. [18] [19]

As Hadoop has the capability of data storage as well as the prcoessing framework as MapReduce also have dozens, sometimes hundreds, of CPUs computational processors in them. When the data scientists can apply R to these predictive models, they can get the power house for computations. R language enables data scientists to take full advantage of the computational capacity of Hadoop. Big data analysis developers can use many R language intensive data analysis tools for refining and extraction of information, such as capturing the signal from the noise, , extracted from the social network graph showing the measured data. For Hadoop execution, including in-database execution, parallelized user code. parallelized algorithms, multi-core processing, multi-threaded execution, memory management and fast math libraries, R can also be scaled up. [19][20]

R language can also be extended for Hadoop execution including in-database execution, parallelized user code, parallelized algorithms, multi-core processing, multi-threaded execution, memory management and fast math libraries.

OPEN SOURCE R ON HADOOP

1. Rhive :It allows the R users to run Hive queries
2. RHadoop : This one is most commonly used R packages for Hadoop which has a very simple interface for running MapReduce jobs using Hadoop streaming.
3. RHIPE : it allows the user to run mappers and reducers in R [18]

VIII. CONCLUSION

The R community is very active in improving the scalability of R. If it is used with Hadoop then there are many more options that aim at parallelizing R across multiple processors or nodes. The main drivers for this work are nothing but a large data usage and the increasing demands of scientific research and high-performance computing.

REFERENCES

[1] Blog on Big Data by Rob Sobers

[2] J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. In OSDI, pages 137–150, 2004

[3] A Book on Big Data Analytics with R and Hadoop, Vignesh Prajapati, Packt Publishing

[4] Paul Torfs & Claudia Brauer, A very Short Description to R, the part of R-website

[5]Source : https://www.rstudio.com/

[6]Source: http://www.vertica.com/distributedr/distributedrtutorial/ - R Tutorial

[7]Feng Wang,Bo Dong,Jie Qiu,Xinhui Li,Jie Yang,Ying Li, "Hadoop High Availability through Metadata Replication" CloudDB '09 Proceedings of the first international workshop on Cloud data management, ACM New York, ISBN 978-1-60558-802-5/09, 2009, Pages 37-44

[8] Daniel Peng and Frank Dabek , "Large-scale Incremental Processing Using Distributed Transactions and Notifications", Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation, USENIX , 2010

[9] An article written by Michael Walker , www.analyticbridge.com/profiles/blogs/percolator-dremel-and-pregel-alternatives-to-hadoop, August 12, 2012

[10]Ricardo: Integrating R and Hadoop, Sudipto Das, Yannis Sismanis, Kevin S. Beyer,Rainer Gemulla Peter J. Haas, John McPherson. ACM, SIGMOD 2010

[11]Tyson Condie, Neil Conway, Peter Alvaro, Joseph M.Hellerstein, Khaled Elemeleegy, Russel Sears, "Map Reduce Online", Proceedings in NSDI'10 Proceedings of the 7th USENIX conference on Networked systems design and implementation, pages 1-14, Oct 9 2009

[12]Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber, "Bigtable: A Distributed Storage System for Structured Data", ACM Transactions on Computer Systems (TOCS), Volume 26 Issue 2, June 2008, Article No.4 , pages1-14, 2006

[13] Revolution Analytics White Paper : Advanced "Big Data" Analytics with R and Hadoop

[14] Jose Pino, Janani Chakkradhari , High level languages for Big Data Analytics , June 2013, Page 1-16

[15] L. Tierney, A. J. Rossini, N. Li, and H. Sevcikova,. SNOW: Simple network of workstations. Tehttp://cran.r-project.org/web/packages/snow/

[16]Hortonworks, Enabling R on Hadoop, July 11 2013.

[17] Joab Jackson, Hadoop gets native R tools for big data analysis, Dec 16 2013

[18] Ross Ihaka and Robert Gentleman,
R: A Language for Data Analysis and Graphics, Journal of Computational and Graphical Statistics, vol.5 No.3, pp 299-314,
Feb 2009

[19] Sukhendu Chakraborty, "Making R Work with Big Data", Oct 23 2013

[20] Saroj Kar, Hadoop Rises and Becomes Stronger with Native R Programming Support, Dec 24 2013