An Approach to mining massive Data

Reena Bharathi
Dept Of Computer Science
Nowrosjee Wadia College Pune

Nitin N Keswani
Dept of Comp Science Nowrosjee
Wadia College
Pune

Siddesh D Shinde
Dept of Comp Science
Nowrosjee Wadia College
Pune

ABSTRACT

Modern internet applications, scientific applications have created a need to manage immense amounts of data quickly. According to a Study, the amount of information created and replicated is forecasted to reach 35 zettabytes (trillion gigabytes) by the end of this decade. The exponentially growing dataset is known as **Big Data.** Big Data is generated by number of sources like Social Networking and Media, Mobile Devices, Internet Transactions, Networked Devices and Sensors

Data mining is the process of extracting interesting, non-trivial, implicit, previously unknown and potentially useful patterns or knowledge from huge amount of data [9]. Traditional mining algorithms are not applicable to Big data as the algorithms are not scalable In many of these types of applications, the data is extremely large and hence there is an ample opportunity to exploit parallelism, in the management & analysis of this type of data. Earlier methods of dealing with massive data were by using the concepts of parallel processing / computing, with a setup of multiple nodes / processors. With the advent of Internet, distributed processing using the powers of multiple servers located on the internet became popular. This led to the development of S/w frameworks, to deal with analysis & management of massive datasets. These s/w frameworks use the concept of a distributed file system, where data & computations on it can be distributed across a large collection of processors.

In this paper, we propose a method for dealing with large data sets, using the concept of distributed file systems and related distributed processing, The Apache HADOOP (HDFS). HDFS is a software framework that supports data-intensive distributed applications and enables applications to work with thousands of nodes and petabytes of data. Hadoop MapReduce [1] is a software framework for distributed processing of large data sets on compute clusters, which enable most of the common calculations on large scale data to be performed on large collections of computers , efficiently & tolerant to h/w failures during computations.

We include in this paper , a case study of a mining application, for mining a large data set (a Email Log) that uses the Apache Hadoop framework for preprocessing the data & converting it into a form, acceptable as input to traditional mining algorithms.

General Terms

Big data, Distributed file systems, Data mining Algorithms

Keywords

Hadoop HDFS, MapReduce, Dendograms, Clusters, Web services, Cloud

1. INTRODUCTION

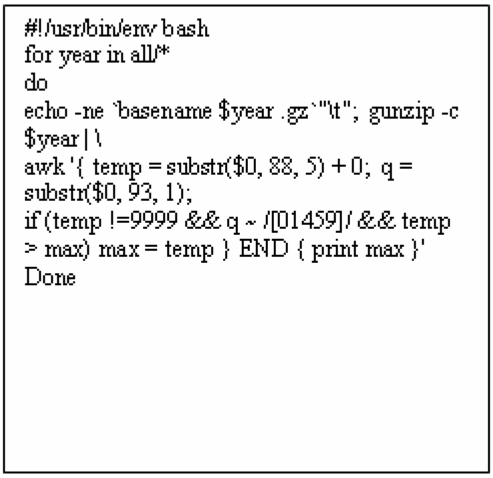
Recent years have witnessed the prevalence of BIG DATA (massive data) in many scientific and commercial applications, such as social networking, Weather data management & analysis, bioinformatics etc. The emergence of BIG data places new challenges for managing and mining this data, as follows:

- More processing and computing power required to handle huge data.
- Applying Mining techniques on such huge data is Time and Space Consuming.
- Larger data hence more work load, thus its distribution is a major headache.
- If distributed in client-server fashion, it requires a good control over involved machines, data Integrity checkups ,optimization of resources etc

Big Data often requires heavy preprocessing, so that only relevant information is extracted out. For example, an application relating to BIG data, is mining Weather data. Weather sensors collecting data every hour at many locations across the globe gather a large volume of log data, which is a good candidate for mining. The data is stored using a line-oriented ASCII format, in which each line is a record. Each record supports a rich set of meteorological elements, many of which are optional or with variable data lengths.

Since there are tens of thousands of weather stations, the whole dataset is relatively small files, where each file contains daily weather information for a year. Analyzing this data to generate the highest recorded global temperature for each year is a time consuming task, if done as a single task, without parallelism.

Different tools are available that can be used to process line-oriented data and one of the classic tool is awk. Figure 1 displays a small awk script to calculate the maximum temperature for each year. [1] The script loops through the compressed year files, first printing the year, and then processing each file using *awk.* The complete run for the century took 42 minutes in one run on a single EC2 High-CPU Extra Large Instance [1]



```
#!/usr/bin/env bash
for year in all/*
do
echo -ne `basename $year .gz`"\t"; gunzip -c
$year | \
awk '{ temp = substr($0, 88, 5) + 0; q =
substr($0, 93, 1);
if (temp !=9999 && q ~ /[01459]/ && temp
> max) max = temp } END { print max }'
Done
```

Figure 1 : An Awk script [1]

To speed up the processing, we need to execute parts of the program in parallel. But then it involves drawbacks like difficulty in dividing work into equal-size pieces, more processing needed while combining the results coming from different processors, limited processing capacity of single machines, should be scalable, reliability issues etc. Hence even though parallel processing is feasible, it involves more complexity. Using a distributed file system framework like APACHE HADOOP helps to care of issues mentioned above. Hadoop enables the development of reliable, scalable, efficient, economical and distributed computing using very simple Java interfaces.

In this paper, we describe how a Hadoop distributed file system framework can be used to preprocess massive data to extract out only relevant information that can then be used in mining process. The entire process of mining massive data is defined a 4 phase program, as follows:

a) The Map phase ➔ Mapping creates a new output list by applying a function to individual elements of an input list , thereby extracts out relevant information from the source data set

b) The sort & shuffle phase ➔ sorts the output list obtained after mapping

c) The Reduce phase ➔ reducing a list iterates over the input values to produce an aggregate value as output. (the preprocessed data, to be used for mining)

d) The Mining phase ➔ Application of a mining algorithm on the reduced data set, to generate interesting information / pattern.

The rest of this paper is organized as follows. Preliminaries are outlined in section 2. The application of HDFS to a case study is outlined in section 3. Section 4 presents the experimental results, of statistical analysis on the data set. Section 5 concludes on this study with a brief discussion on the further extension of this research work.

2. PRELIMINARIES

2.1 The Hadoop File System (HDFS):

Hadoop [2] is an open source implementation of the MapReduce parallel processing framework. Hadoop hides the details of parallel processing, thereby allow the developers to focus on their computational problem, rather than on the complexities related to parallelization. The Hadoop framework is optimized to run on a cluster of commodity servers that are horizontally scalable: adding more servers adds more compute and storage capacity.

The Hadoop framework thus provides the following advantages:

- Fault tolerance by detecting faults and applying quick, automatic recovery.
- Data access via MapReduce programming model.
- Scalability to reliably store and process large amounts of data.
-
- Economy by distributing data and processing across clusters of commodity Personal Computers.
- Efficiency by distributing data and logic to process it in parallel on nodes where data is located.
- Reliability through replication of data
- Automatically redeploying processing logic in the event of failures.

A HDFS [fig 2] cluster has two types of nodes, working in a master-worker pattern: a namenode (master) and a number of data nodes (worker nodes). The master node manages the filesystem namespace & it knows the data nodes on which all the blocks for a given file are located. The datanodes are the workers of the file system. They store and retrieve blocks as per the request from clients or the name nodes and also report back the storage information with respect to the list of blocks to the namenode.

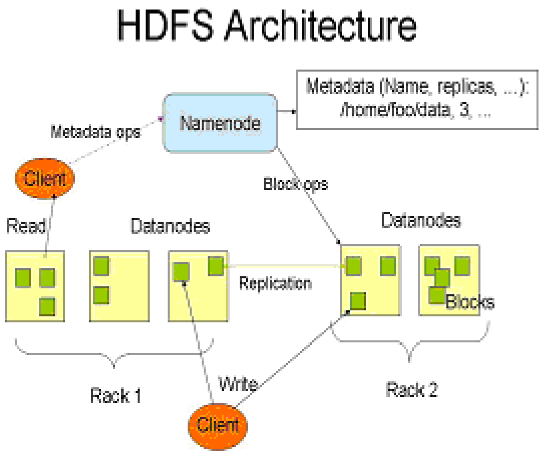


Figure 2 : Hadoop Architecture [4]

2.2 MapReduce Programming Tool:

MapReduce is a programming model for data processing. Hadoop supports MapReduce programs written in various languages like Java, Rails, and Python etc. MapReduce programs are, by nature, parallel, thereby putting very large-scale data analysis into any setup with enough machines at their disposal. It works in two phases: the map phase and the reduce phase. Each phase has a Key-value pair as input and output., whose types are chosen by the programmer. The map function and the reduce function are programmer specified. An input file is fed to the Map function, which produces a set of Key-value pairs . In the Reduce phase, all records with same key are collected and fed to the same reduce process, which produces a final set of data values. The whole data flow is illustrated in fig 3 [1].

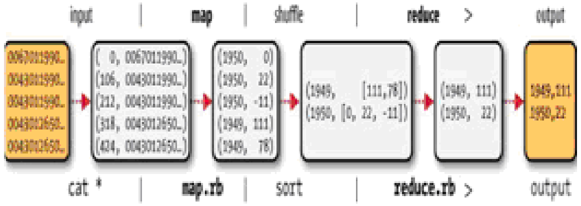


Figure 3 : Map- Reduce logical data flow [1]

The MapReduce task processes are horizontally scalable; adding more servers adds more computational power.

3. A Case Study

Email logs are a useful resource for research in fields like social network analysis, textual analysis, fraud detection etc. The Enron email dataset [3] was made public by the Federal Energy Regulatory Commission during its investigation. It contains all kind of emails personal and official.

Cohen from CMU has put up the dataset on the web for researchers (http://www-2.cs.cmu.edu/~enron/).

In this paper we have considered the Enron email dataset containing 252,759 messages from 151 employees distributed in around 3000 user defined folders, as the source data set, for our statistical analysis. . The dataset contains the folder information for each of the 151 employees. Each message present in the folders contains the senders and the receiver email address, date and time, subject, body, text and some other email specific technical details.

Fig 4 shows the workflow of our case study project, implemented using the Hadoop framework. The Email dataset is moved onto the HDFS. The Map phase is then applied on the Email dataset to produce a set of key-value pairs, where the key is the email subject and the corresponding value is the list of participants.

In the Reducer phase, all mails with the same subject are collected, and fed to the same reduce process, to produce the final set of data values. The final set of data values obtained after reduction is a very large set and hence it's not possible to extract information by just browsing through it.

Mining algorithms are applied on this final data set to extract hidden information / patterns. We applied clustering algorithms and analyzed the statistics of the dataset, to verify its appropriateness and the following section 4 describes the Descriptive data mining results thus obtained.

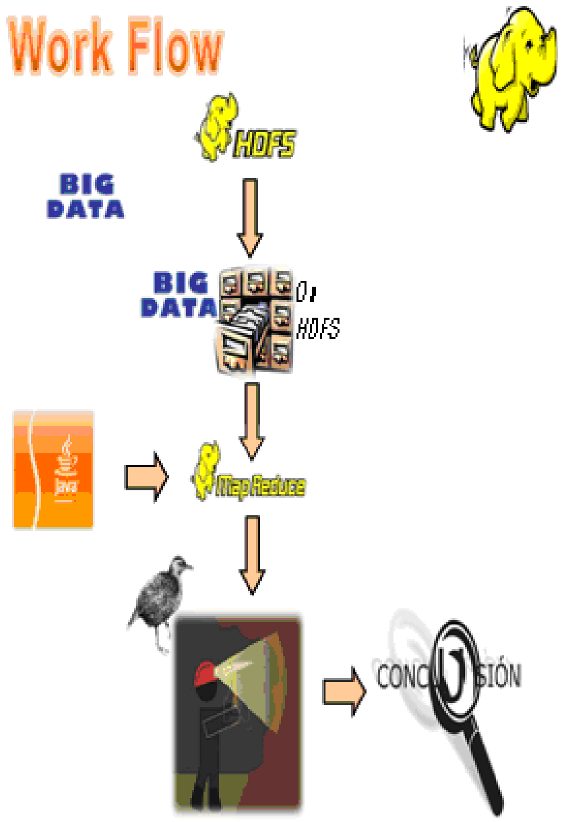


Figure 4 : Workflow of Hadoop MapReduce process

4. DESCRIPTIVE DATA MINING

We applied some descriptive data mining techniques such as summarization and clustering on the reduced data set. These unsupervised learning techniques capture the intrinsic structure of the data and present it in a visual form for human comprehension

4.1 Clustering

Clustering explores similarities found in the dataset to build a set of clusters. Hierarchical clustering methods group data objects into a tree of clusters with different set of clusters at different levels off hierarchy. A tree structure called dendrogram presents the outcome of a hierarchical clustering algorithm

Using R package, we applied the Hierarchical clustering algorithm and plotted a cluster dendogram [Fig 5] that depicts the similarity between the mail subjects. The similarity between any two mails is defined as the ratio of Number of common participants to total number of participants in the mail subjects. The distance or dissimilarity is computed as 1- similarity. This dissimilarity matrix is fed as input to hierarchical clustering algorithm, to generate a Dendogram, as shown in Fig 5.

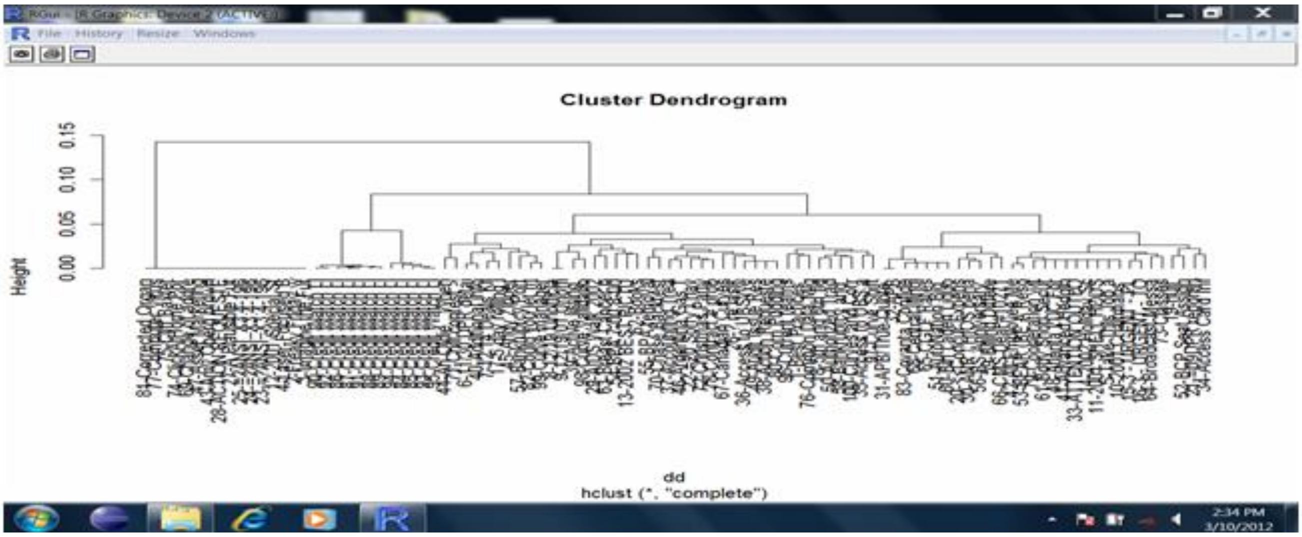


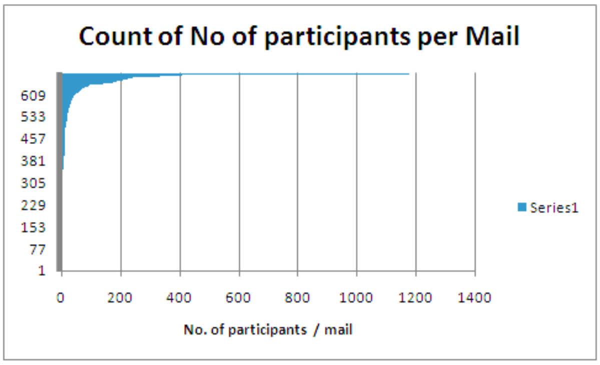
Figure 5 : A cluster Dendogram

Analyzing the Dendogram, it's clear that, most of the mails have a different set of participants list, since the number of clusters at the lowest level is many. Very few mails have a good similarity with each other, with respect to the participant list.

4.2 Statistical Summarization

Summarization derives representative summary type of information from the dataset that sums up the data characteristics. We used the MS- Excel to plot charts showing the Number of participants per subject. Analyzing the charts, it's clear that most mails (approx. 150 mails from 693 mails) have only one participant in its list [Fig 6]. Out of the rest, the number of participants varied upto around 400, except for one mail that had 1180 participants in its list. This mail is clearly shown as an outlier in the chart [Fig 7], which means that the mail may be a general mail sent to the company's customers. This statistical information, along with some Textual analysis of the email contents (the email body), can be used to mine useful information or patterns from the Email data set.

Fig 6 : Plot of Number of Participants per mail



by defining them as web services on the Hadoop framework Thus we plan to build a set of web services on a scientific cloud ,using the Hadoop framework, that will make it easy &

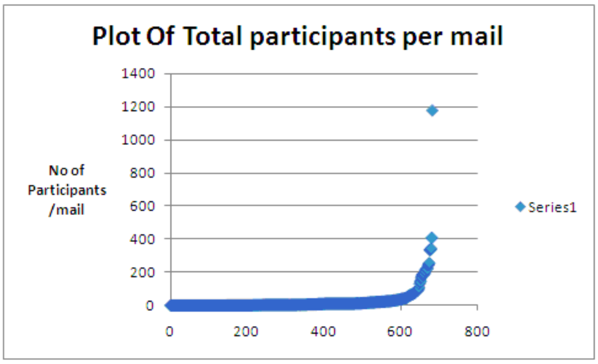


Fig 7 A Scatter Plot of No of Participants / mail

5. GENERALIZATION OF CONCEPT

Big data refers to data sets that are too large to be processed and analyzed by traditional IT technologies. Scientific applications like Weather sensor data, GIS based data Genetic applications etc some of the candidates that generate huge data (BIG DATA). These applications require features like scaling, data partitioning, replication, and data consistency, to solve their data processing needs. Scientific Clouds provide services to address the Big data storage and processing issues in scientific applications.

In this paper, we have described how massive data sets can be reduced and used for mining applications, using the Hadoop framework and the MapReduce programming model. A Hadoop framework in the cloud can be configured to run on large cluster servers [2]. The convenience of storing and processing Big data in the cloud means that the data is in Hadoop clusters. The data can be stored in the HDFS at collection time, processed using MapReduce, and delivered to consumers without being stored in another filesystem or database.

Mining algorithms on massive datasets can be parallelize

cost effective to store, process and extract information from massive volumes of data , as shown is Fig 8

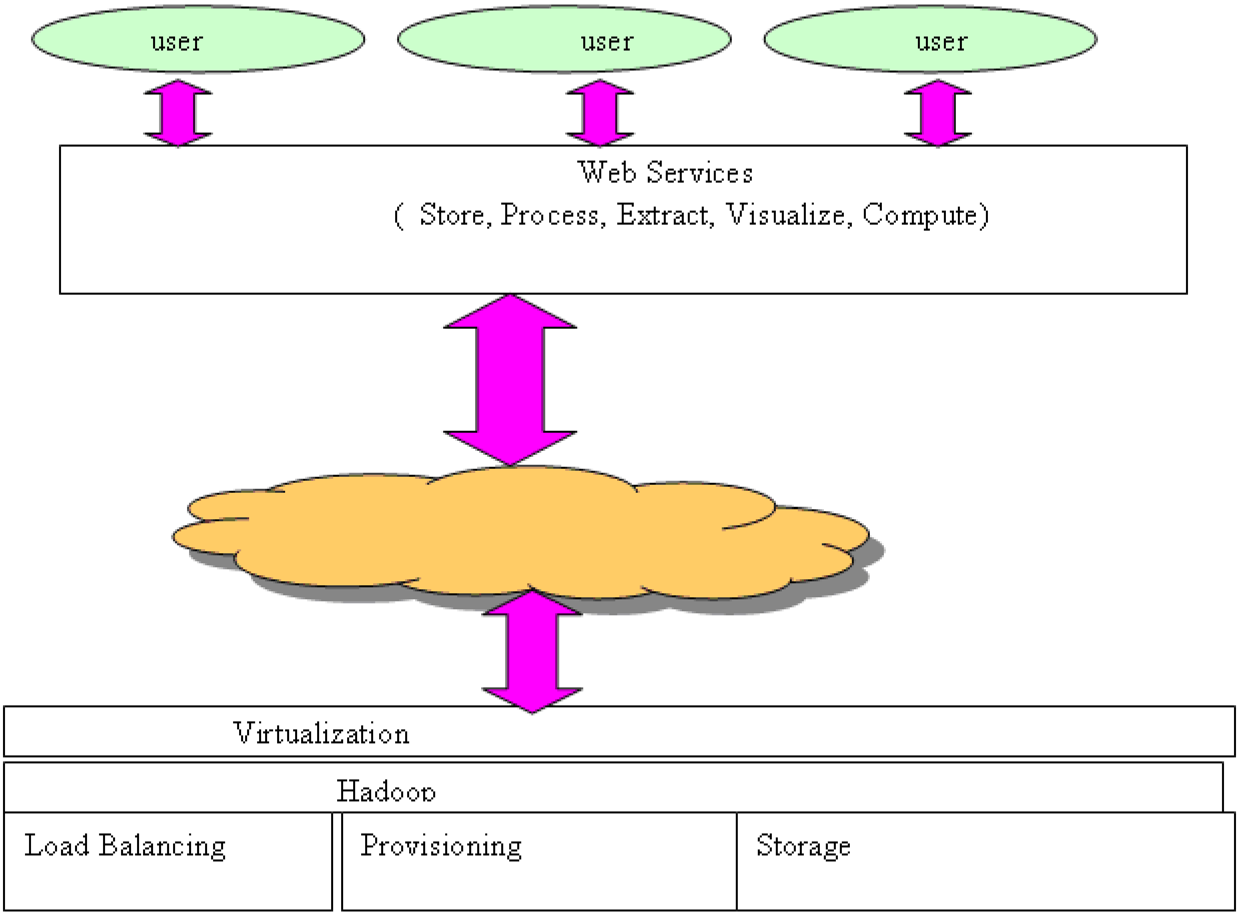


Figure 8 : Computing on the cloud

6. ACKNOWLEDGMENTS

Our thanks to Dr. S.C Shirwaikar, Head, Dept of Computer Science, who provided us all the necessary guidance, for the completion of this research work.

7. REFERENCES

[1] Tom White, 2011, Hadoop The Definitive Guide 2nd Edition 2010, O'Reilly.

[2] Brian F Cooper, Eric Baldeschwieler, Rodrigo Fonseca, James J Kistler, P.P.S. Narayan, Chuck Neerdaels, Toby Negrin, Raghu Ramakrishnan, Adam Silberstein, Utkarsh Srivastava, Raymie Stata, 2009, IEEE, Building a Cloud for Yahoo!

[3] Jitesh Shetty, Jafar Adibi, The Enron Email Dataset Database schema and Brief Statistical Report.

[4] Druba Borthakur, 2009 Microsoft Research Seattle, Hadoop Architecture and its usage at Facebook

[5] Andrew Pavlo, Erik Paulson, Alexander Rasin ,Daniel J. Abadi , David J. DeWitt, Samuel Madden, Michael Stonebraker, Copyright 2009 ACM, A Comparison of Approaches to Large-Scale Data Analysis

[6] Siyang Dai, Jinxiong Tan, Zhi Zhang, Zeyang YU, Shuai Yuan, MR Language Reference Manual

[7] Jimmy Lin, Chris Dyer, 2010, Data-Intensive Text Processing with MapReduce

[8] Dell | Cloudera Solution for Apache Hadoop Deployment Guide, www.dell.com

[9] S. Jiawei Han and Micheline Kamber. 2006. Data mining Concepts and techniques: Morgan/Kauffman publishers.