A Survey on Big Data Analysis Techniques

Himanshu Rathod[1] Tarulata Chauhan[2]
[1, 2]Department of Computer Engineering
[1, 2]L.J.I.T, Ahmedabad, Gujarat

Abstract—There is a growing trend of applications that ought to handle huge information. However, analysing huge information may be a terribly difficult drawback nowadays. For such data many techniques can be considered. The technologies like Grid Computing, Volunteering Computing, and RDBMS can be considered as potential techniques to handle such data. We have a still in growing phase Hadoop Tool to handle such data also. We will do a survey on all this techniques to find a potential technique to manage and work with Big Data.

Key words: big data, big data analysis, hadoop;

I. INTRODUCTION

Large-scale data and its analysis are at the centre of modern research and enterprise. These data are known as Big Data. The facts and figures of such data are developed from online transactions, internet messages, videos, audios, images, click streams, logs, posts, search queries, wellbeing notes, social networking interactions, science facts and figures, sensors and wireless phones and their submissions . They are retained in databases grow hugely and become tough to capture, pattern, store, manage, share, investigate and visualize by usual database software tools. [1]

5 Exabyte (1018 bytes) of facts and figures were conceived by human until 2003. Today this amount of data is created in two days. In 2012, digital world of data was expanded to 2.72 zettabytes (1021 bytes). It is forecast to double every two years, reaching about 8 zettabytes of facts and figures by 2015. IBM indicates that every day 2.5 exabytes of facts and figures created furthermore 90% of the facts and figures made in last two years. An individual computer retains about 500 gigabytes (109 bytes), so it would need about 20 billion PCs to shop all of the world's facts and figures. In the past, human genome decryption method takes roughly 10 years, now not more than a week. Multimedia facts and figures have large-scale weight on internet backbone traffic and are anticipated to boost 70% by 2013. Only Google has got more than one million servers around the worlds. There have been 6 billion mobile subscriptions in the world and every day 10 billion text messages are sent. By the year 2020, 50 billion devices will be attached to systems and the internet. [1]

II. RELATED WORK

There are many techniques to handle Big data. Here we will discuss some Big Data storage techniques as well as some of the Big data analysis techniques. Their brief study is as mentioned below.

A. Hadoop

Hadoop is an open source project hosted by Apache Software Foundation. It consists of many small sub projects Which belong to the category of infrastructure for distributed computing. Hadoop mainly consists of:

1) File System (The Hadoop File System)
2) Programming Paradigm (Map Reduce)

The other subprojects provide complementary services or they are building on the core to add higher-level abstractions. There exist many problems in dealing with storage of large amount of data.

Though the storage capacities of the drives have increased massively but the rate of reading data from them hasn't shown that considerable improvement. The reading process takes large amount of time and the process of writing is also slower. This time can be reduced by reading from multiple disks at once. Only using one hundredth of a disk may seem wasteful. But if there are one hundred datasets, each of which is one terabyte and providing shared access to them is also a solution. [2]

There occur many problems also with using many pieces of hardware as it increases the chances of failure. This can be avoided by Replication i.e. creating redundant copies of the same data at different devices so that in case of failure the copy of the data is available. [2]

The main problem is of combining the data being read from different devices. Many a methods are available in distributed computing to handle this problem but still it is quite challenging. All the problems discussed are easily handled by Hadoop. The problem of failure is handled by the Hadoop Distributed File System and problem of combining data is handled by Map reduce programming Paradigm. Map Reduce basically reduces the problem of disk reads and writes by providing a programming model dealing in computation with keys and values. [2]

Hadoop thus provides: a reliable shared storage and analysis system. The storage is provided by HDFS and analysis by Map Reduce. [2]

B. Hadoop Components in detail

1) Hadoop Distributed File System:

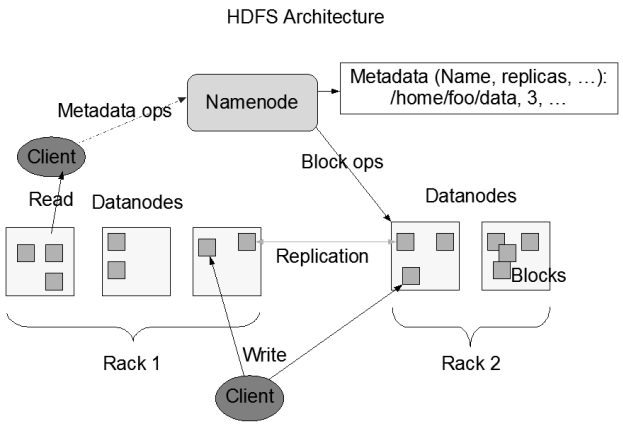

Fig. 1: HDFS Architecture [4]

Hadoop comes with a distributed File System called HDFS, which stands for Hadoop Distributed File System. HDFS is a File System designed for storing very large files with streaming data access patterns, running on clusters on commodity hardware. HDFS block size is much larger than that of normal file system i.e. 64 MB by default. The reason for this large size of blocks is to reduce the number of disk seeks. [4]

A HDFS cluster has two types of nodes i.e. namenode (the master) and number of datanodes (workers). The name node manages the file system namespace, maintains the file system tree and the metadata for all the files and directories in the tree. The datanode stores and retrieve blocks as per the instructions of clients or the namenode. The data retrieved is reported back to the namenode with lists of blocks that they are storing. Without the namenode it is not possible to access the file. So it becomes very important to make name node resilient to failure. [4]

These are areas where HDFS is not a good fit: Low-latency data access, Lots of small file, multiple writers and arbitrary file modifications. [4]

2) MapReduce:

MapReduce is the programming paradigm allowing massive scalability. The MapReduce basically performs two different tasks i.e. Map Task and Reduce Task. A map-reduce computation executes as follows:

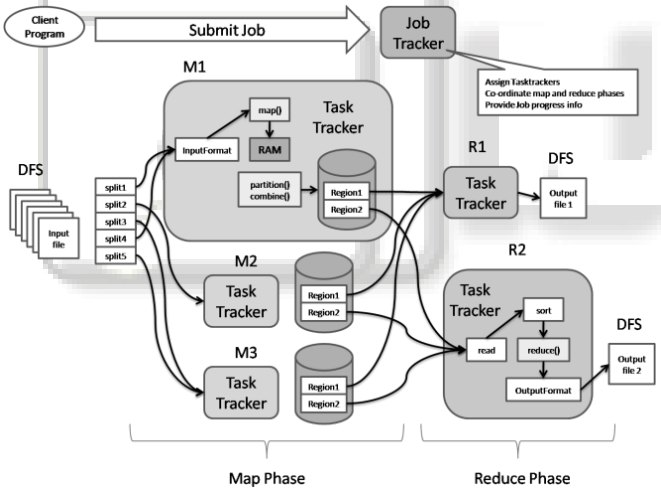

Fig. 2: MapReduce Architecture

Map tasks are given input from distributed file system. The map tasks produce a sequence of key-value pairs from the input and this is done according to the code written for map function. These value generated are collected by master controller and are sorted by key and divided among reduce tasks. The sorting basically assures that the same key values ends with the same reduce tasks. The Reduce tasks combine all the values associated with a key working with one key at a time. Again the combination process depends on the code written for reduce job. [5]

The Master controller process and some number of worker processes at different compute nodes are forked by the user. Worker handles map tasks (MAP WORKER) and reduce tasks (REDUCE WORKER) but not both. [5]

The Master controller creates some number of maps and reduces tasks which are usually decided by the user program. The tasks are assigned to the worker nodes by the master controller. Track of the status of each Map and Reduce task (idle, executing at a particular Worker or completed) is kept by the Master Process. On the completion of the work assigned the worker process reports to the master and master reassigns it with some task.

The failure of a compute node is detected by the master as it periodically pings the worker nodes. All the Map tasks assigned to that node are restarted even if it had completed and this is due to the fact that the results of that computation would be available on that node only for the reduce tasks. The status of each of these Map tasks is set to idle by Master. These get scheduled by Master on a Worker only when one becomes available. The Master must also inform each Reduce task that the location of its input from that Map task has changed. [5]

III. COMPARISON OF HADOOP TECHNIQUE WITH OTHER SYSTEM TECHNIQUES

A. Grid Computing Tools:

The approach in Grid computing includes the distribution of work across a cluster and they are having a common shared File system hosted by SAN. The jobs here are mainly compute intensive and thus it suits well to them unlike as in case of Big data where access to larger volume of data as network bandwidth is the main bottleneck and the compute nodes start becoming idle. Map Reduce component of Hadoop here plays an important role by making use of the Data Locality property where it collocates the data with the compute node itself so that the data access is fast.[2,6]

Grid computing basically makes a use of the API's such as message passing Interface (MPI). Though it provides great control to the user, the user needs to control the mechanism for handling the data flow. On the other hand Map Reduce operates only at the higher level where the data flow is implicit and the programmer just thinks in terms of key and value pairs. Coordination of the jobs on large distributed systems is always challenging. Map Reduce handles this problem easily as it is based on shared-nothing architecture i.e. the tasks are independent of each other. The implementation of Map Reduce itself detects the failed tasks and reschedules them on healthy machines. Thus the order in which the tasks run hardly matters from programmer's point of view. But in case of MPI, an explicit management of check pointing and recovery system needs to be done by the program. This gives more control to the programmer but makes them more difficult to write. [2, 6]

B. Comparison with Volunteer Computing Technique:

In Volunteer computing work is broken down into chunks called work units which are sent on computers across the world to be analysed. After the completion of the analysis the results are sent back to the server and the client is assigned with another work unit. In order to assure accuracy, each work unit is sent to three different machines and the result is accepted if at least two of them match. This concept of Volunteer Computing makes it look like MapReduce. But there exists a big difference between the two the tasks in case of Volunteer. Computing is basically CPU intensive. This tasks makes these tasks suited to be distributed across computers as transfer of work unit time is less than the time required for the computation whereas in case of MapReduce

is designed to run jobs that last minutes or hours on trusted, dedicated hardware running in a single data centre with very high aggregate bandwidth interconnects.[2,6]

C. Comparison with RDBMS:

The traditional database deals with data size in range of Gigabytes as compared to MapReduce dealing in petabytes. The Scaling in case of MapReduce is linear as compared to that of traditional database. In fact the RDBMS differs structurally, in updating, and access techniques from MapReduce. Comparison of RDBMS over some general properties of big data is as shown below. [2, 6]

	Traditional RDBMS	*MapReduce*
Data Size	Gigabytes	Petabytes
Access	Interactive and batch	Batch
Updates	Read and write many times	Write once, read many times
Structure	Static schema	Dynamic Schema
Integrity	High	Low
Scaling	Nonlinear	Linear

Table. 1: RDBMS compared to MapReduce [6]

IV. CONCLUSION AND FUTURE WORK

From the above study and survey we can conclude that Hadoop is possibly one of the best solutions to maintain the Big Data. We studied other techniques such as Grid Computing tools, Volunteering Computing and RDBMS techniques. We learnt that Hadoop is capable enough to handle such amount of data and analyze such data.
In future we would like to carry out unstructured data of logs with the help of Hadoop. We would like to carry out analysis more efficiently and quickly.

REFRENCES

[1] Sagiroglu, S.; Sinanc, D., "Big data: A review," Collaboration Technologies and Systems (CTS), 2013 International Conference on , vol., no., pp.42,47, 20-24 May 2013
[2] Katal, A.; Wazid, M.; Goudar, R.H., "Big data: Issues, challenges, tools and Good practices," Contemporary Computing (IC3), 2013 Sixth International Conference on , vol., no., pp.404,409, 8-10 Aug. 2013
[3] Nandimath, J.; Banerjee, E.; Patil, A.; Kakade, P.; Vaidya, S., "Big data analysis using Apache Hadoop," Information Reuse and Integration (IRI), 2013 IEEE 14th International Conference on , vol., no., pp.700,703, 14-16 Aug. 2013
[4] Apache Hadoop (2013). HDFS Architecture Guide [Online]. Available: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.ht
[5] Apache Hadoop (2013). Hadoop Map/Reduce Tutorial [Online]. Available: http://hadoop.apache.org/docs/r0.18.3/mapred_tutorial.html
[6] Tom White (2013). Inkling for Web [Online]. Available: ttps://www.inkling.com/read/hadoop-definitive-guide-tom-white-3rd/chapter-1/comparison-with-other-systems

