International Journal of Research in Advance Engineering, (IJRAE)
Vol. 2, Issue 3, May-June-2016, Available at: **www.knowledgecuddle.com/index.php/IJRAE**

Finding Insights & Hadoop Cluster Performance Analysis over Census Dataset Using Big-Data Analytics

Dharmendra Agawane [1], Rohit Pawar [2], Pavankumar Purohit [3], Gangadhar Agre [4]
Guide: Prof. P B Jawade
[2]rohitpawar95@hotmail.com
[1234] Information Technology Department, Govt. College of Engineering. Karad
Saidapur, Dist: Satara, Maharashtra, India.

***Abstract*:** Big-data is a very important resource in today's world, if utilized properly. But for utilizing this big data it is required to analyse that data to find out some interesting facts out of it. The increase in the Five V's of the data i.e. Velocity, Volume, Variety, Veracity and Value have made the processing of the data more and more complex and this change is bringing more and more challenges in this field of Data Processing.
In such a case to get some useful information out from such a large variety and volume of data we need to use the concepts like DATA MINING and CLUSTERING. In our project we would be finding different insights from US census Dataset which will help us to understand the dataset more easily and conclude inferences from it. For this purpose i.e. for finding the insights from a large data set we would be working with the framework called "Hadoop", which is a tool for processing big-data with minimum time and with more accuracy.
Then we would be analysing the performance of the Hadoop cluster both single node and multi-node while working with different number of nodes with different block size in addition with replication factor.

Keyword: **Hadoop, Map-Reduce, Insights, Single node Hadoop Cluster, Multi node Hadoop Cluster, Replication, Block size.**

I. INTRODUCTION

Now a day, large amounts of data is generated continuously from various applications which include YouTube, Business Computing, Internet, Facebook, and Twitter.

With the Increase in size of the data every day, there is a need to handle, manage and analyse for the Business Applications and future prediction. To handle such large volume of semi-structured and unstructured data, Google's Map Reduce technique has proven to be an efficient tool. Map Reduce (proposed in 2004), is an efficient programmable framework for handling large data with single node and multinode cluster.

Hadoop, developed by Apache Foundation, is an open source framework written in java language that allows distributed processing of large datasets on single node and multimode cluster. It was developed by Doug Cutting and Mike Cafarella in Yahoo in the year 2005 and it was handover to Apache for further development. Hadoop is built to handle terabytes and petabytes of semi-structured and unstructured data. Hadoop is the answer to many questions generated by the Big Data.

In this paper, we analyse the performance of Hadoop in a Single Node Cluster, by finding insights from US census Data set (using different block sizes & Replication factor) and then analysing its performance on multi node.

II. Hadoop Architecture

Hadoop is an open source implementation of the Map-Reduce programming, developed by Yahoo and supported by Apache Software Foundation. Hadoop is fault tolerant, scalable, flexible and distributed system for data storage and data processing.

INDEX COPERNICUS INTERNATIONAL
ICV 2014: 53.07
Standardized Value: 5.71
eISSN: 2395-1648

There are two main components of Hadoop: Hadoop HDFS (Hadoop distributed File System) and Hadoop Map Reduce. Hadoop HDFS is for the storage of data i.e. hadoop file storage and Hadoop Map Reduce for processing, parallel computation of data and retrieval of data. Map Reduce is considered the heart of the Hadoop system which performs the parallel processing over large datasets generally in size of terabytes and petabytes. Hadoop is based on batch processing and handles large unstructured and semi-structured data as compared to traditional relational database systems which works on the structured data.

i. HDFS : Hadoop distributed File System

HDFS is "Hadoop Distributed File System" implemented by Yahoo based on Google File System. As its name implies, it is a distributed, fault tolerant, reliable file system. HDFS can be seen as Master/Slave architecture which contains NameNode, DataNode and Secondary NameNode. Name Node is the master node which controls all the DataNodes and handles all the file system operations.

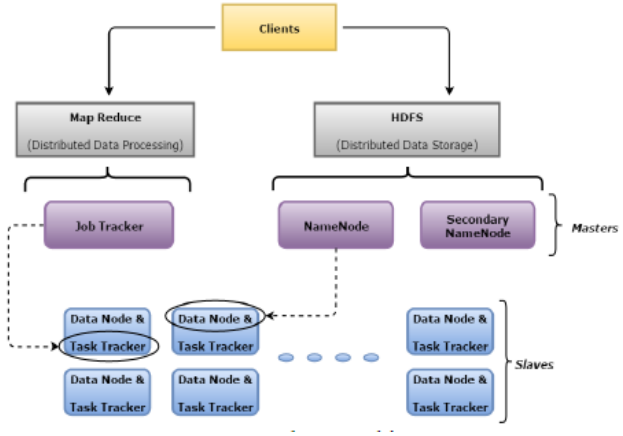


Figure 1: Hadoop Architecture [1]

DataNodes are the slave nodes which perform the actual working like block operations. There is also Single Secondary NameNode in HDFS which acts like the backup node of NameNode. The Master partitions the data into blocks which are then stored on the DataNodes in a cluster. With the default replication factor of three, HDFS places the copies of the block to the local node, second and third DataNodes. This block replication is used for avoidance of data loss in case of DataNode failure which can be adjusted as per our convenience and need. The default block size in HDFS is defined as 64 MB which can also be increased if required.

ii. MapReduce

MapReduce is a technique invented by Google for processing data in parallel. The MapReduce works on the large datasets stored in the HDFS of the Hadoop. Job initialization, task assignment, status update, task execution, progress and Job submission are the activities carried by the MapReduce. All these activities are handled by JobTracker and carried by TaskTracker.

The operation takes place in two phases: Map Phase and Reduce Phase. In Map phase data is partitioned into chunks giving the output in <key,value> pairs i.e. tuple. These pairs i.e. tuples are then passed to the Reducer phase which combines all the outputs and produce the single output.

III. Proposed System Architecture:

The proposed system architecture consist of two main sub-systems. The first one is for user interaction and providing a job to the Hadoop. While the second subsystem takes the input from the HDFS and works in visualizing the results. In the first subsystem the raw data which is in a CSV format is first loaded in the HDFS and the user choice is taken. According to the user choice the particular map-reduce job is called and the working on the dataset is done for

INDEX COPERNICUS INTERNATIONAL
ICV 2014: 53.07
Standardized Value: 5.71
eISSN: 2395-1648

finding out the particular results. The results are produced and then stored back to the HDFS. During the same process the system configuration and performance of the Hadoop is stored in HDFS as a pdf file.

In the second subsystem, i.e. the visualization part; the Tom-cat server takes the input from the HDFS and then the visualization process is carried forward. The visualized part is then viewed in the web-browser. While working on the multi-node cluster all the working of the system remains same except the Hadoop map-reduce working. In the multi-node cluster the job of map-reduce is divided into number of small jobs on different nodes; the results of which are again combined to form the final result and this result is then stored back to the HDFS for further use.

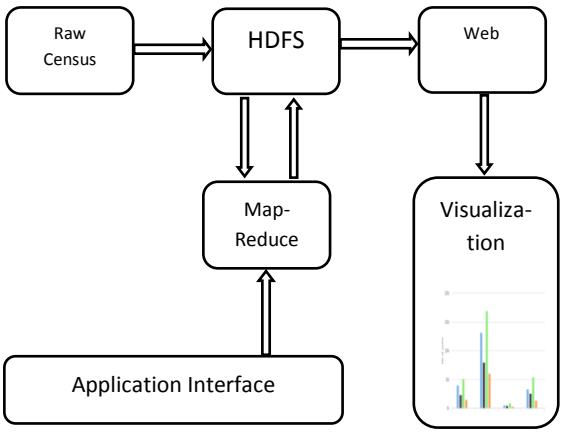


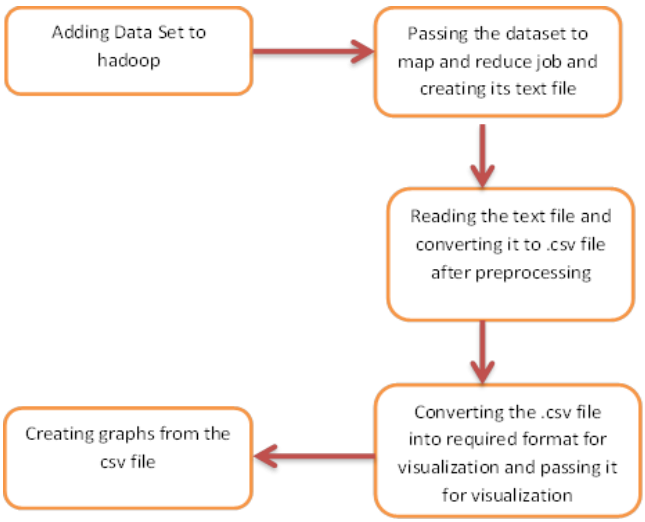
Figure 2: Proposed System Architecture

If the multi-node cluster is used with the small dataset then it may lead to the increase in the overall working time due to the additional time of splitting and joining the job between multiple nodes, while if the same is used with the larger dataset it reduces the overall operation time because of parallel and distributed computing.

IV. Work Flow Diagram:

In Data-set processing the raw data i.e. the .CSV file of the census which is stored in the HDFS is passed to the map-and-reduce phase of the Hadoop. Before going to the map phase the user choice for the particular insight is taken from the Application Interface and then accordingly the particular map-reduce job is called for processing.

After the reduced data is stored in the HDFS output folder, the file for the visualization is prepared. For this the output file is converted in suitable format (for required for visualizing the output data).



ICV 2014: 53.07
Standardized Value: 5.71
eISSN: 2395-1648

Figure 3: Work Flow Diagram

After this conversion the generated file is directly stored in the "web-apps" folder of apache tomcat where the process of visualization takes place.

V. Data Visualization (Dynamic Graphs):

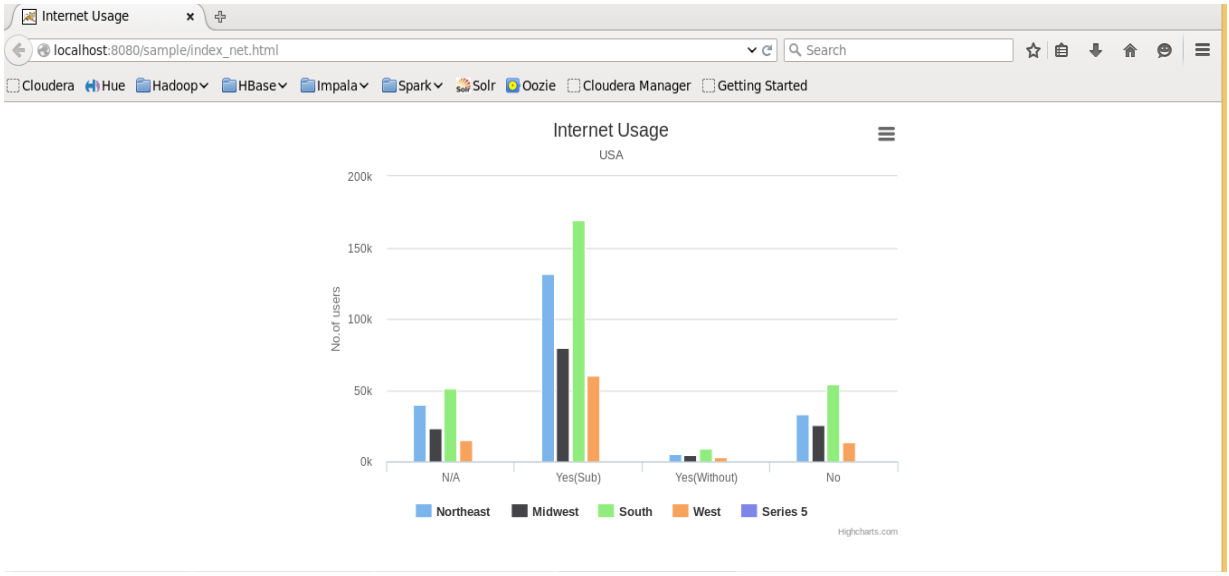


Figure 4: Data Visualization

This is the basic structure of how the project works. But the main work done using this project is the performance analysis of the Hadoop cluster on multi-node cluster.

The performance analysis of the Hadoop Cluster is:

Table1: performance analysis of the Hadoop Cluster

Block Size	Replication				
64	2	Number of Nodes	1	2	3
		Time for Processing	58	44	39
64	1	Number of Nodes	1	2	3
		Time for Processing	58	48	36
128	2	Number of Nodes	1	2	3
		Time for Processing	53	39	37

ICV 2014: 53.07
Standardized Value: 5.71
eISSN: 2395-1648

From the different readings taken during the execution of the project on different nodes with different replication we come across some fact such as: As the no of nodes in the Hadoop Cluster increases, execution time decreases provided that the data set is large enough else it may increase.

Also we understood that the replication factor affects the execution time.

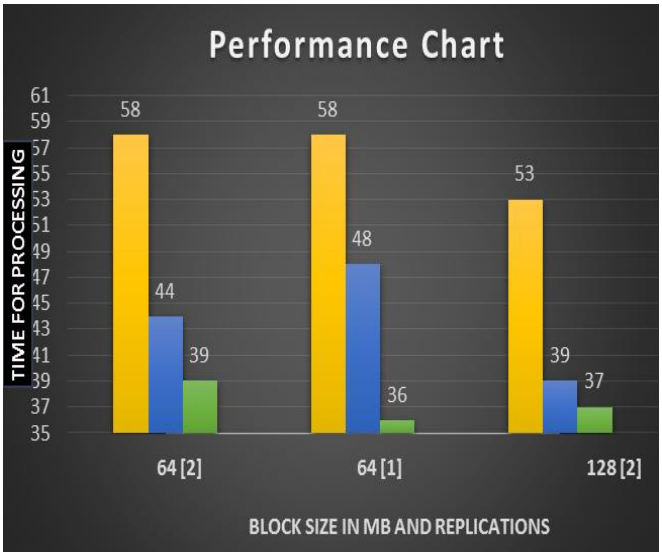


Figure 4: Performance Analysis chart

VI. Technical Specification:

Hardware Requirements

System clock speed : 1.70-2.40 GHz
Hard Disk : Minimum 30 GB(Virtual Image).
RAM : 1.Heterogeneous Environment 4-8GB (for master) and 2GB (for slaves).
2. Homogeneous Environment 4GB onwards (master/slave)

Software Requirements

Operating System : CentOS (CDH5) / Ubuntu.
Technology : Hadoop with multimode cluster, Java, JavaScript, Highchart.
FileSystem : HDFS.

VII. Conclusion:

In this paper, we have analysed and observed the results of MapReduce application on different Hadoop clusters using different block sizes and with different replication factors. In hadoop clusters, it has been concluded that a threshold exists below which adding more nodes does not result in performance enhancement of the cluster. But after that value, with increasing number of DataNodes, the Hadoop cluster performance can be enhanced.

REFERENCES

[1]. Ruchi Mittal and Ruhi Bagga, "Performance Analysis of Multi-Node Hadoop Clusters using Amazon EC2 Instances", International Journal of Science and Research (IJSR) ISSN (Online): 2319-7064

INDEX COPERNICUS INTERNATIONAL
ICV 2014: 53.07
Standardized Value: 5.71
eISSN: 2395-1648

[2]. Jiong xie & Jshu Yin, "Improving Map Reduce Performance through Data placement in Heterogeneous Hadoop Cluster", Department of Computer Science and Software Engineering Auburn University, Auburn, AL 36849-5347

[3]. Dr.Kiran Jyoti & Mrs.Bhawna Gupta, "Big Data Analytics with Hadoop to Analyze Targeted Attacks on Enterprise Data by", (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3), 2014, 3867-3870

[4]. Zaheer Khan & Ashiq Anjum, "Cloud based Big Data Analytics for Smart Future Cities", Utility and Cloud Computing (UCC), 2013 IEEE/ACM 6th International Conference.

[5]. Vishal S. Patil & Pravin D. Soni, "Hadoop Skelton and fault tolerance in Hadoop Cluster", International Journal of Application or Innovation in Engineering & Management (IJAIEM), Volume 2, Issue 2, February 2013, ISSN 2319 – 4847.

[6]. Xindong Wu & Xingquan Zhu, "Data Mining with Big Data", ieee transactions on knowledge and data engineering, vol. 26, no. 1, january 2014.

[7]. P. Prabhu and N. Anbazhagan, "Improving the Performance of K-means Clustering for High Dimensional Data Set", International Journal on Computer Science and Engineering (IJCSE)

[8]. Apache Hadoop. http://hadoop.apache.org/

[9]. www.michael-noll.com/tutorials/running-hadoop-on-ubuntu-linux-single-node-cluster

[10]. www.michael-noll.com/tutorials/running-hadoop-on-ubuntu-linux-multi-node-cluster

[11]. http://www.edupristine.com/courses/big-data-hadoop-program/big-data-hadoop-course/

ICV 2014: 53.07
Standardized Value: 5.71

eISSN: 2395-1648