Implementation of Word Count- Hadoop Framework with Map Reduce Algorithm

Keerthi.Bangari[#1], Sujitha.Meduri[*2], Ch.CY.Rao[#3]

[#]*Assistant Professors, Department of Computer Science and Engineering, Geethanjali College of Engineering and Technology(Autonomous)*
Hyderabad, India

Abstract *— In the technological world there are number of technologies which are generating a large amount of data day by day that leads to formation of a technology called Big Data. Big Data deals with the large and unstructured data that can be computationally analysed to reveal the trends and patterns of a data. In this paper the basic program called Word Count Map Reduce program executed in apache hadoop with a single node setup. Altering in input files and reducing the number of tasks that makes the changes in execution of a program. The aim of this paper is running the Word Count program with different parameters.*

Keywords—*Word Count Program, Apache Hadoop, Map Reduce, Big Data, Parameters*

I. INTRODUCTION

Big Data term defines the collection of large data sets, where the data is in structured and unstructured formats. Structured data can present in the form of table format. So the data can be easy to analyze and processed by using data mining tools. Unstructured data refers the data does not have any table format i.e. it does not have any structure. So it is not resided in any traditional databases. Several challenges are encountered in big data while processing, storing and analyzing the data. To fast process the large volume of data within the short period of time, a tool is required which is called Hadoop. Hadoop is open source software which is developed by Apache for reliable, scalable and distributed computing. Apache Hadoop is a framework it uses simple programming models for distributed processing of large volume of data sets through the clusters of computers. It is designed to set up from single server to group of machines, each system offers the storage and local computation.

HDFS and Map Reduce are the two major concepts of Hadoop. Both the Map Reduce and Hadoop are related to distributed computation. Basically Hadoop architecture is same as distributed master slave architecture I.e. in master slave only one system acts as master and remaining systems acts like servers. The use of Hadoop Distributed File System is for distributed and storage for computational capabilities. The purpose of Hadoop is for partitioning the data and it perform parallel computing for large data sets. In Map Reduce master schedules the work on the slave nodes. And the HDFS master is responsible partitioning and computing the data from slaves and it keeps track of the data where it is located.

II. RELATED WORK

In Map Reduce word count is one of the primary program. For easy understanding the simple map reduce performance model was done by using word count program. In this paper Map Reduce performance model was evaluated by changing the size of input file and modifying in map spilt granularity. By altering the size of input file the map reduces performance model can be identified. If there is increase in size of input file then the execution time of word count program can increase. If the size of input file is small then the execution time of word count program will be less. This paper explains the map reduce performance model can be evaluated by changing the size of the input file based on it the variations can be done in word count program executions.

III. INSTALLATION OF HADOOP

Before executing the word count program in Hadoop. Hadoop is an open source software framework it was written in java. Hadoop is used to run the applications on large clusters. First install the Hadoop framework in single node. Before installing the Hadoop framework the java software need to be installed. In .bashrc file set the java and Hadoop path. And configure the files called core-site.xml,hdfs-site.xml,mapred-site.xml,yarn-site.xml which is present in Hadoop folder. To run the Hadoop first format the name node as

Hadoop name node –format

Hadoop namenode format it used to start the namenode in Hadoop

After starting namenode in Hadoop the dfs and yarn daemons are used to run

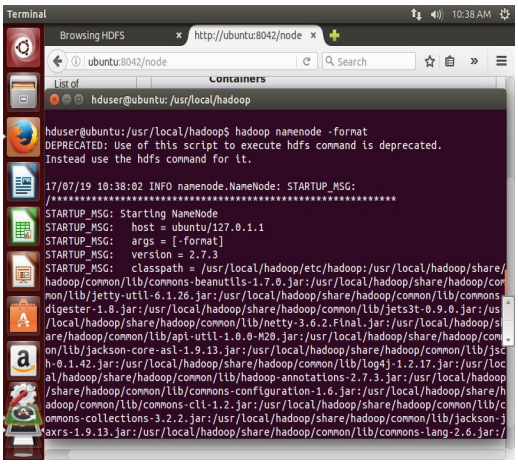

Figure.1 Screenshot for starting of namenode.

After starting the namenode start all Hadoop daemons like dfs.sh,yarn.sh by using the command

Start-all.sh

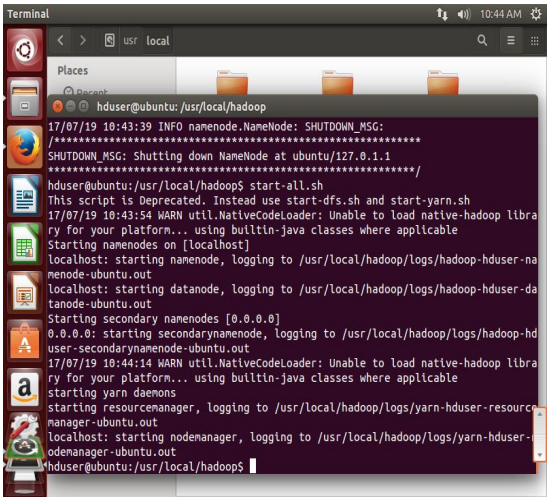

Figure.2 Screenshot for starting all the nodes in Hadoop.

To stop all the daemons like dfs.sh,yarn.sh stop-all.sh command can be used

Stop-all.sh

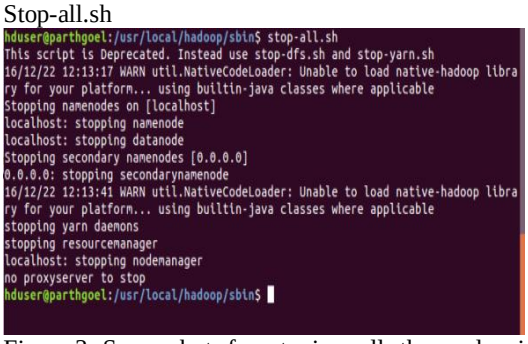

Figure.3 Screenshot for stoping all the nodes in Hadoop.

Jps is a command it is used to check if all the daemons in Hadoop are running are not.

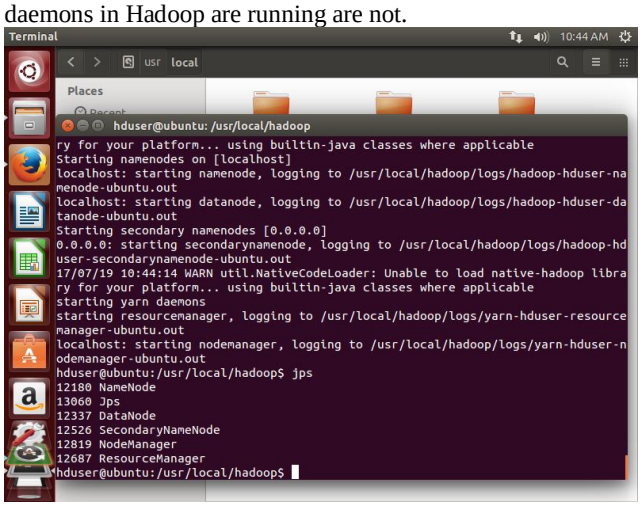

Figure.4 Screenshot for currently running nodes in Hadoop.

After completing the installation steps in hadoop login to browser by using the following links to run the programs in hadoop.

http:localhost:8088.

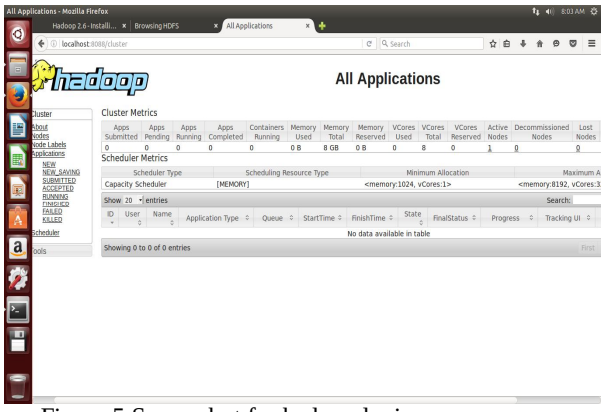

Figure.5 Screenshot for hadoop login page .

http:localhost:50070.

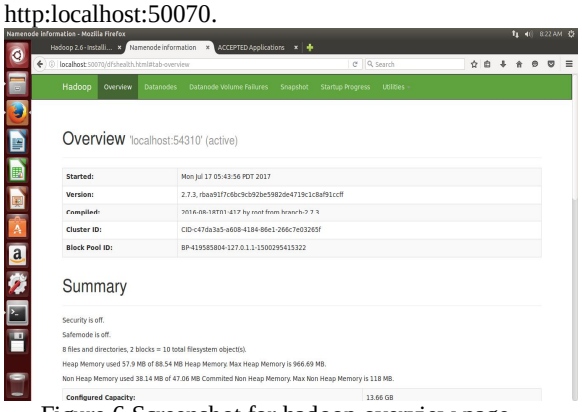

Figure.6 Screenshot for hadoop overview page..

IV. EXECUTION OF WORD COUNT PROGRAM USING MAP REDUCE TECHNIQUE IN SINGLE NODE HADOOP

Word Count is the basic program in Hadoop. The source code of Hadoop downloaded in Apache site .In this paper the Hadoop-2.7.3 version was used. Word Count Program present in the examples of Hadoop it is in the form of jar file. Word Count Program reads the input file called text file and it counts the occurrences of each word which is present in the text file. Both the input and output files are text files. Each line has words the word count program counts the text file words and the words how repeatedly it occurred it is separated by tap. The program accommodates map and reducer interfaces to implement the map and reduce tasks.

A. *Mapper*

Mapper used to map the group of input keys or value pairs to a group of intermediate keys. Whatever the given input keys may maps to zero or it can map to output keys. In Hadoop map reduce framework based on Input Split the Input Format for the job can be generated. All the Intermediate keys are combined with output keys are consequently grouped by Hadoop framework and it can be passed to reducers to identify the final output. The Mapper outputs can be sorted and it is partitioned per reducer. The number of reduced tasks for the job is same as total number of partitions done by the Mapper.

B. *Reducer*

Reducer can reduce the set of intermediate keys which share the keys to small group of values. Job.setNumReduceTasks(int) command used to reduce the number of tasks in a job. Reducer had three phases:Sort,Shuffle and Reduce. In sorted phase output is sorted and returns the value. In shuffle phase output is collected from the mapper phase.

V. STEPS TO RUN THE WORD COUNT PROGRAM IN HADOOP WITH DIFFERENT TEXT FILES.

A. *Starting all hadoop daemons*

Before running the program in hadoop first start all daemons by using the following commands.

Hadoop namenode –format
Start-all.sh
Stop-all.sh

B. *Running the programs in hadoop by using the following commands.*

To execute any programs in hadoop, create the input and output directories in hdfs by using the following commands.

-mkdir -p /user/hduser

bin/hdfs dfs -put /home/mapreduce/desktop/data input

bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.3.jar wordcount input output

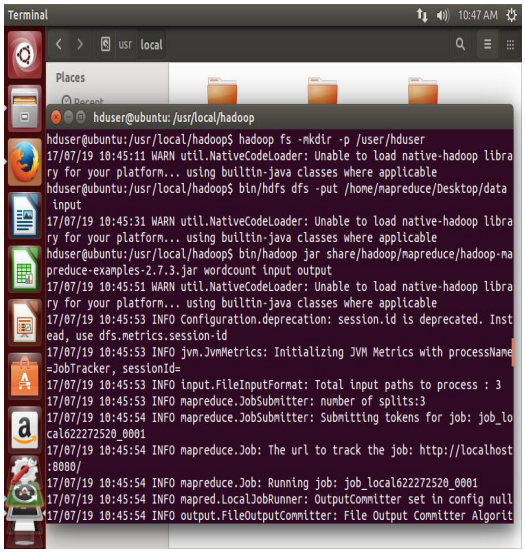

Figure.7 Screenshot for running the word count program in hadoop.

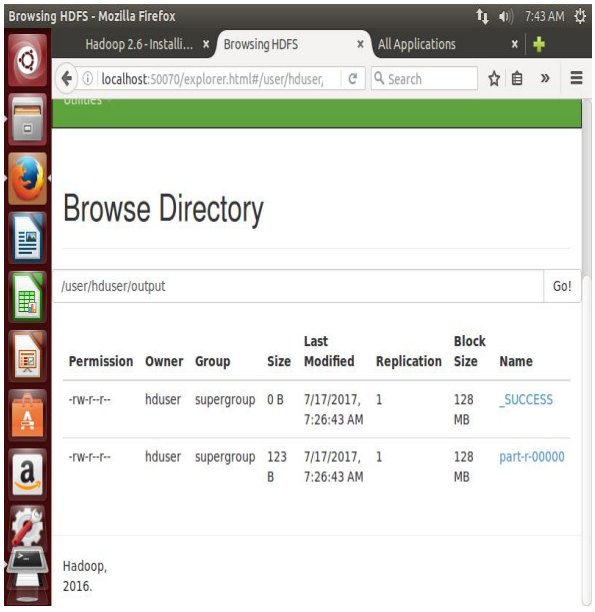

Figure.8 Screenshot for storage of input files in hadoop.

bin/hadoop fs –cat */output

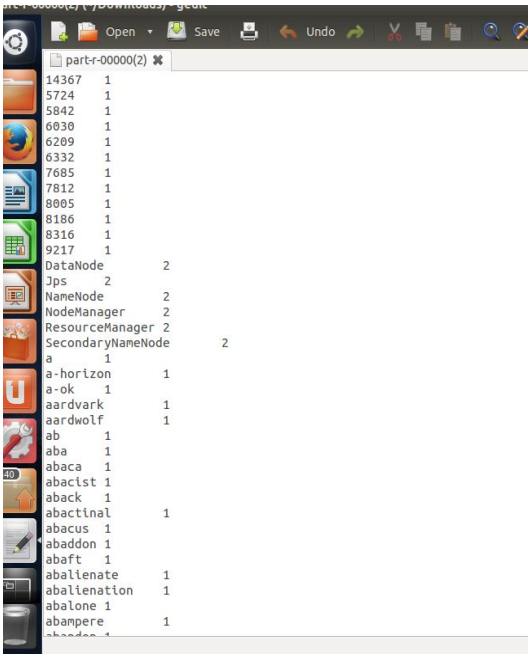

Figure.9 Screenshot for output of word count program in Hadoop.

VI. RESULTS

In VMware environment the ubuntu-14.04 operating system were installed. It is a single node set up. In ubuntu environment Hadoop-2.7.3 framework was install by using various commands. After installing Hadoop the word count program was executed by inserting text file in it.

In the execution of a word count program first the input text files need to be store in hdfs. For different executions of word count program the different input text files can be stored in hdfs based on it execution time of word count program can vary.

Size of input file	Execution time in seconds
100MB	150
300MB	255
1GB	600

Table 1:Table for input file size and its execution time

Figure 10: Graph for input file size varying in job execution time.

X axis –Size of input file
Y-axis-Execution time of a job.

The above bar graph shows the different execution time for different input file sizes.

VII. CONCLUSION AND FUTURE WORK

In ubuntu operating system in single node set up Hadoop framework was installed. In Hadoop word count program executed with different input file sizes where in each file execution time will be vary. In this paper the author observed by increasing the size of file there will be increasing in the execution time of a file. But in future work it will be extended to multi node cluster.

REFERENCES

[1] http://hadoop.apache.org/ ,Apache Hadoop
[2] Maurya, M., & Mahajan, S. (2012, October). Performance analysis of MapReduce programs on Hadoop cluster. In Information and Communication Technologies (WICT), 2012 World Congress On (pp. 505-510). IEEE.
[3] Yang, X. and Sun, J., 2011, September. An analytical performance model of mapreduce. In Cloud Computing and Intelligence Systems (CCIS), 2011 IEEE International Conference on (pp. 306-310). IEEE.M. Wegmuller, J. P. von der Weid, P. Oberson, and N. Gisin, "High resolution fiber distributed measurements with coherent OFDR," in *Proc. ECOC'00*, 2000, paper 11.3.4, p. 109.
[4] Zaharia, Matei, et al. "Improving MapReduce performance in heterogeneous environments." Osdi. Vol. 8. No. 4. 2008. (2002) The IEEE website. [Online]. Available: http://www.ieee.org/
[5] https://www.tutorialspoint.com/hadoop/ ,Map Reduce tutorial
[6] *Chavan, Vibhavari, and Rajesh N. Phursule. "Survey paper on big data." Int. J. Comput. Sci. Inf. Technol 5, no. 6 (2014): 7932-7939.*
[7] Blazhievsky, S., 2013. Introduction to Hadoop, MapReduce and HDFS for Big Data Applications. SNIA Education.
[8] Arora, Suman, and Dr Madhu Goel. "Survey paper on Scheduling in Hadoop." International Journal of Advanced Research in Computer Science and Software Engineering 4.5 (2014).
[9] Tan, Jian, Xiaoqiao Meng, and Li Zhang. "Performance analysis of coupling scheduler for mapreduce/hadoop." INFOCOM, 2012 Proceedings IEEE. IEEE, 2012.