IJCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Volume-3, Issue-6 E-ISSN: 2347-2693

Performance Analysis of Hadoop with Pseudo-Distributed Mode on Different Machines

Ruchi Mittal[1*] **and Ruhi Bagga**[2]

[1*,2]*Department of Computer Science & Engg.,Punjab Technical University, India*
mittal.ruchi29@gmail.com, ruchi.bagga86@gmail.com

www.ijcseonline.org

Received: Jun /05/2015 Revised: Jun/09/2015 Accepted: Jun/18/2015 Published: Jun/30/ 2015

Abstract— Data cannot be managed by the traditional database management systems when it comes in a large amount. So there comes the Big Data. Hadoop and MapReduce are the solution to handle, manage and analyze Big Data. Hadoop is an open source implementation of MapReduce programming paradigm which is a parallel distributed programming model for handling large data intensive applications. In this paper, we present our experimental work done on Hadoop with pseudo-distributed mode on different machines and analyze the time taken by Hadoop to perform the same operations on different machines.

Keywords— Big Data; Hadoop; MapReduce; Pseudo-distributed Mode; Distributed Programming

I. Introduction

Nowadays, data is generated continuously by the use of various applications such as business computing, internet, social media (e.g. Facebook, Twitter) and scientific research. With the growing size of data every day, the need to handle, manage and analyse that data is also growing. To handle such large volume of unstructured data, MapReduce has been proven an efficient technique.

MapReduce is an efficient programmable framework for handling data in a parallel manner in a cluster of many systems. This model was first proposed by Google in 2004. Hadoop is an open source implementation of the MapReduce framework. There is large volume of information residing in the form of unstructured data like e-mails, PDF files, text files, spreadsheets etc. on the hard drives in the companies. Hadoop is built to handle terabytes and petabytes of a. It is used to handle large datasets over extensive applications. It answers many questions generated by the challenges of big data. Hadoop is a distributed software solution for a cluster of machines with data in different formats

II. HADOOP ARCHITECTURE

Hadoop is an open source implementation of the MapReduce programming paradigm supported by Apache Software Foundation. It is a scalable, fault tolerant, flexible and distributed system for data storage and processing. There are two main components of Hadoop- Hadoop HDFS and Hadoop MapReduce. Hadoop HDFS is for the storage of data and Hadoop MapReduce for processing and retrieval of data. MapReduce 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 data as compared to traditional relational database systems which works on the structured data only.

A. *MapReduce*

MapReduce is the programmable framework for data in parallel in a cluster. The applications are written using MapReduce programming which act on the large datasets stored in the HDFS in Hadoop. Job submission, job initialization, task assignment, task execution, progress and status update and all other activities related to the job completion are handled by MapReduce. In this, all the activities are managed by the JobTracker and are executed by the TaskTracker which are the main components of the MapReduce.

In this processing is carried out in two different phases namely Map Phase and Reduce Phase. In map phase, the input is splitted into small size chunks which are processed in parallel. The output of this phase are the <key, Value> pairs which are given to the reducer which combines all the outputs to produce a single output.

JobTracker and TaskTracker are the main components of MapReduce which exploits the master/slave architecture.

- JobTracker is the master to the TaskTracker. It schedules and coordinates all the jobs submitted by the client and also handles the task distribution to the TaskTracker. JobTracker and TaskTracker use heartbeat messages to communicate with each other. When a job is submitted by the client, the JobTracker communicates with the NameNode to locate the data required for processing which then submits the job to the different TaskTrackers for processing. TaskTrackers periodically send heartbeat messages to the JobTracker to ensure that they are alive and doing the task allocated. If JobTracker doesn’t receive a message from a TaskTracker for a particular period of time, it then considers that node to be dead and reallocates the task allocated to that TaskTracker to some other alive TaskTracker.

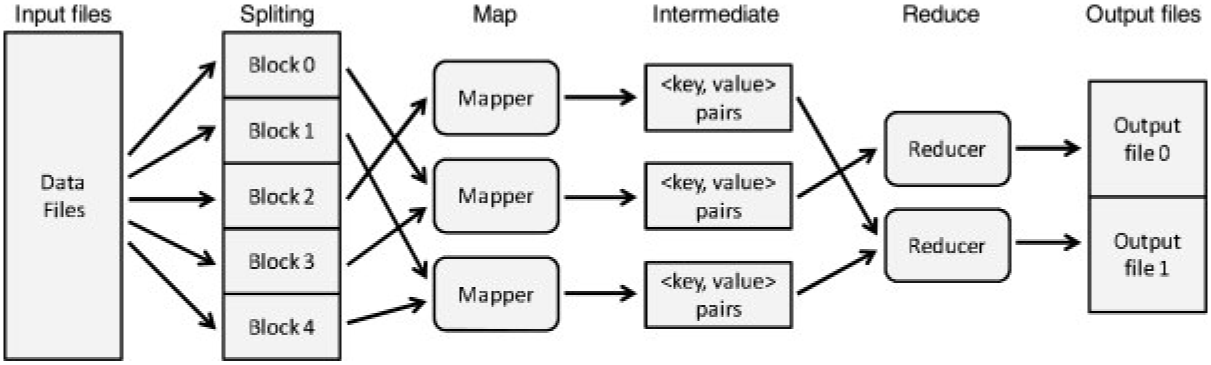

Fig.1. An overview of MapReduce model[9]

- TaskTracker receives the job from the JobTracker and breaks them into the map and reduce tasks. It executes the tasks and report the status update to the JobTracker with the output.

B. HDFS

HDFS is the Hadoop Distributed File System implemented by Yahoo based on Google File System. As its name implies, it is a distributed, reliable, fault tolerant file system. HDFS can be seen as master/slave architecture which contains NameNode, DataNode and Secondary NameNode. NameNode is the master node which controls all the DataNodes and handles all the file system operations like managing all the namespaces, block mapping, breaking a file into blocks. DataNodes are the slave nodes which perform the actual working like block operations like storage of data blocks, replications over the blocks etc. There is also Secondary NameNode in HDFS which acts like the housekeeping node of NameNode.

HDFS partitions the data into blocks which are stored on the DataNodes. With the replication factor of three, HDFS places the first copy of the block to the local node, second to the other DataNode of the local rack and the last copy to the different node in the different rack. The default block size in HDFS is defined as 64 MB which can also be increased if required. NameNode, DataNode and Secondary NameNode are the main components of HDFS, role of each of which is discussed as below:

- NameNode is the master node of HDFS which communicates with the HDFS Client. It maps the whole data of the cluster to different DataNodes. It also keep track of all the transactions carried out in the cluster. When the NameNode is down, the whole cluster is down. It stores and manages the metadata about the complete file system of the cluster in a file named fsimage.
- DataNodes are the slave nodes of the HDFS which performs the actual operations on the request submitted by the client to the NameNode. DataNodes communicate with the NameNode by sending heartbeat messages. The blocks are stored on the DataNodes.
- Secondary NameNode acts like the housekeeping node of the NameNode. It checks the file system for changes periodically and merge them into the fsimage file which contains the metadata about the file system. On the failure of NameNode, the information about the blocks can be recovered from it.

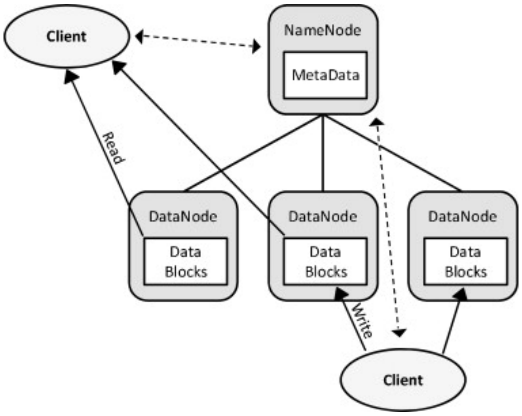

Fig.2. An overview of HDFS read and write.[9]

C. Hadoop Modes

Hadoop can be installed in one of the three different modes:

1) The Standalone Mode: In this mode, all hadoop daemons run under a single java process. You donot need to configure anything. This is the default mode and is recommended for testing purpose.

2) The pseudo-distributed mode: Hadoop is configured for all the nodes in this mode. A separate java virtual machine(JVM) is configured for each node like a cluster on a single system.

3) The full distributed mode: In this mode, each daemon runs on different machine. Different machines are configured for hadoop components.

III. Installation of Hadoop Pseudo-Distributed Mode

A. Prequisites Required

For the installation of Hadoop over Ubuntu with pseudo distributed mode, we first need the following pre-requisites:

- Java is the main requirement for the installation of Hadoop as Hadoop requires java for its working. Java can be installed by writing the following command onto the command prompt.

  ```
  apt-get install default-jdk
  ```

- The next main requirement is the configuration of the ssh server. This is required by the master to communicate with its slave nodes. So we need ssh server in our system for communication between the nodes. For this we need to generate a public/private key pair. This generated key will be sent to all the nodes using which they all can communicate with each other. The password less key can be generated using the following commands:

  ```
  ssh-keygen -t rsa -P ''
  ```

  ```
  cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
  ```

B. Steps of Hadoop Installation

- Download and install Hadoop. This can be done form the Apache Software Foundation. After downloading the Hadoop package, we need to extract the package.
- Add paths to the .bashrc. In this we have to add the $JAVA_HOME and $HADOOP_HOME variables to the .bashrc file and update it. By doing this, every time when we will open a shell, Hadoop's home directory will be opened. We need not to set it again and again.
- Configuring Hadoop Environment. In this, we need to set the java home directory in the env.sh script file. By doing this, the Hadoop will get the value of JAVA_HOME for its working every time it is started. We also need to disable the IPV6 as Hadoop doesn't work with it well.
- Configuring the .xml files (core-site.xml, hdfs-site.xml, mapred-site.xml and yarn-site.xml). This will configure the Hadoop configuration files. In core-site.xml file, we define the URI of the NameNode. In the hdfs-site.xml file, we specify the host directories of the NameNode and DataNode. Before editing this file, two directories needs to be created one of which will contain NameNode and the other will contain the DataNode. For each host, this file needs to be set. The mapred-site.xml file contains the information about the framework used for the MapReduce. The yarn-site.xml file contain the configuration properties required by the MapReduce while starting up.
- After setting all the configuration files, the NameNode needs to be formatted for starting Hadoop.
- At the end launch Hadoop daemons to check its working performance.

IV. Experimental Results

This section describes the performance of Hadoop with pseudo distributed mode on different machines.

Machine		Machine Configuration		
		Processor	*Memory*	*Disk*
Machine 1	Toshiba Satellite C50	Intel Corei3-3110M CPU @2.40GHz, 64 bit	2GB	500GB
Machine 2	Sony VAIO	Intel Core2Duo CPU T6600@2.20GHz, 32 bit	4GB	500GB
Machine 3	HP G60 Notebook	Pentium Dual Core CPU T4300@2.10GHz , 64 bit	3GB	300GB

TABLE I. MACHINE SPECIFICATIONS

A. Environment

The above table specifies the configuration of different machines on which the experiment is carried out to measure the performance of Hadoop with pseudo distributed mode. The used Hadoop version on these machines is Hadoop-0.18.0 which is the stable Hadoop version.

B. Results

WordCount is the function which we run on different Hadoop machines with pseudo distributed mode to evaluate their performance in terms of time taken in seconds to complete the job. WordCount is a MapReduce application used to count the number of times a word appears in the input file. We have chosen the WordCount function as this can be used on different files of different sizes. File size taken are from 10 MB to 200 MB. Depending upon these sizes, performance is evaluated for different machines.

Table 2 shows the result of the WordCount executed on different machines. The results are the average execution time in seconds taken in different rounds.

File Size (MB)	Results(in seconds)		
	Machine 1	*Machine 2*	*Machine 3*
10	22.63	30.20	43.15

File Size (MB)	Results(in seconds)		
	Machine 1	*Machine 2*	*Machine 3*
20	33.58	39.95	59.38
40	53.93	74.15	101.66
80	103.90	145.23	187.74
120	155.74	207.84	272.41
160	201.79	273.35	349.65
200	234.68	314.60	424.08

TABLE II. RESULTS OF WORDCOUNT JOB ON DIFFERENT MACHINES WITH 1GB RAM MEMORY

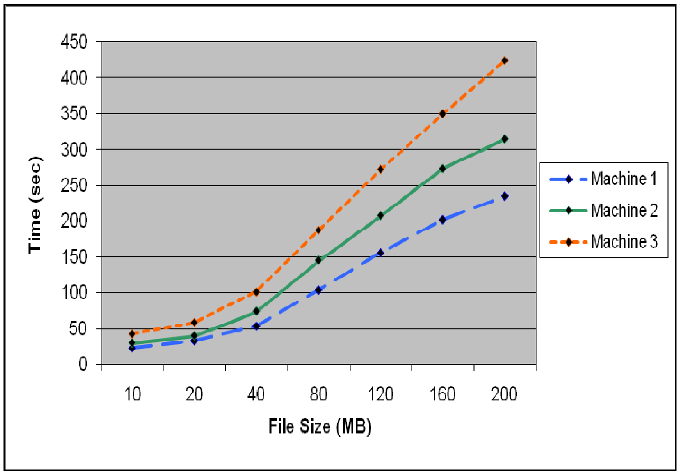

Fig.3. Graph Showing the performance of 3 machines with 1GB RAM

File Size (MB)	Results(in seconds)		
	Machine 1	*Machine 2*	*Machine 3*
10	24.69	28.30	47.43
20	31.74	44.28	69.45
40	56.76	77.23	113.14
80	104.64	152.25	212.44
120	155.06	231.84	315.95
160	222.62	269.40	382.05
200	255.31	331.90	407.92

TABLE III. RESULTS OF WORDCOUNT JOB ON DIFFERENT MACHINES WITH 2GB RAM MEMORY

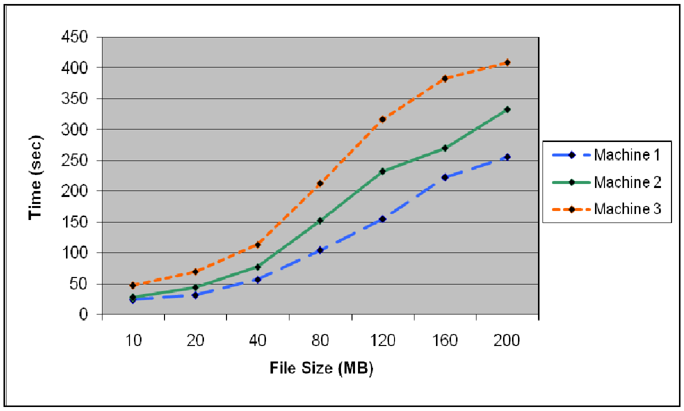

Fig.4. Graph showing the performance of 3 machines with 2GB RAM

Fig. 3 and 4 shows the performance of WordCount application on different machines with Hadoop installed with pseudo distributed mode in terms of their execution time in second on different file sizes. According to the experiments performed, when the file size is up to 120 MB, reduction process starts after the completion of mapping process. But when file size is 160 MB, reduction starts as mapping reaches up to 70% to reduce the overall execution time. As the file size increases, reduction starts along with mapping so that the overall execution time doesn't get increased too much.

V. CONCLUSION

In this paper, we have concluded that Hadoop starts reduction along with the mapping process so that the time taken to process large data files doesn't get increased too much. As Hadoop is used to process large data files, so it has this formulation to handle such large files and to process large data in time. So Hadoop is the best model for handling large, unstructured data.

VI. FUTURE SCOPE

This paper represents the experiment carried out on different machines with Hadoop with pseudo-distributed mode. In future, the work will be carried out on Hadoop with full distributed mode and its performance analysis will be carried out.

ACKNOWLEDGMENT

I would like to thank Er. Ruhi Bagga, AP (CSE Department), Rayat Bahra Group of Institutes, Patiala Campus for having faith in me and allowing me to work with all the terms and conditions and advising me time to time about this work.

REFERENCES

[1] Xuelian Lin, Zide Meng, Chuan Xu, Meng Wang,"A Pratical Performance Model for Hadoop MapReduce", in proc. Of the 2012 IEEE International Conference on Cluster Computing Workshops,ISBN: 978-1-4673-2893-7,Page No (231-239), Sept 24-28,2012.

[2] M. Maurya, S. Mahajan,"Performance Analysis of MapReduce Programs on Hadoop Cluster", in proc. of 2012 World Congress on Information and Communication Technologies, ISBN:978-1-4673-4806-5,Page No (505-510), Oct 30-Nov 2,2012.

[3] M. Ishii, Jungkyu Han, H. Mankino,"Design and Performance Evaluation for Hadoop Clusters on Virtualized Environment", in proc. of 2103 International Conference on Information Networking, E-ISBN:978-1-4673-5741-8, Page No (244-249), Jan 28-30,2013.

[4] Han Jungkyu, M. Ishii, H. Makino,"A Hadoop Performance Model For Multi-Rack Clusters",in proc. of 2013 5th International Conference on Computer Science and Information Technology, Page No (265-274), Mar 27-27,2013.

[5] Zhuoyao Zhang, Ludmila Cherksova, Boon Thau Loo,"Performance Modeling od MapReduce Jobs in Heterogeneous Cloud Environments", in proc. of the 2013 IEEE Sixth International Conference on Cloud Computing, ISBN: 978-0-7695-5028-2, Page No (839-846), June 28- July 3,2013.

[6] J. Nandimath, E. Banerjee, A.Patil, P. Kakade, "Big Data Analysis using Apache Hadoop", in proc. of 2013 IEEE 14th International Conference on Information Reuse and Intergration, Page No (700-703), Aug 14-16,2013.

[7] A. Pal, K.Jain, P.Agarwal,S.Agarwal, "A Performance Analysis of MapReduce Task With Large Number of Files Dataset in Big Data Using Hadoop", in proc. of 2014 Fourth International Conference on Communication Systems and Network Technologies, ISBN: 978-1-4799-3069-2, Page No (587-591), Apr 07-09,2014.

[8] Invanilton Polato, Reginaldo Re, Alfredo Goldman, Fabio Kon, "A Comprehensive view of Hadoop Research- A Systematic Literature Review", Elsevier- Journal of Network and Computer Applications,Volume-46,Page No (1-25), Aug 2014.

[9] Chia-Wei Lee, Kuang-Yu Hsieh Sun-Yuan Hsieh , Hung-Chang Hsiao,"A Dynamic Data Placement Strategy for Hadoop in Heterogeneous Environments ", Elsevier-Big Data Research, Volume-1, Page No (14-22), Aug 2014.

[10] D. Dev, R. Patgiri, "Performance Evaluation of HDFS in Big Data Management", in proc. of 2014 International Conference on High Performance Computing and Applications,ISBN: 978-1-4799-5957-0, Page No (1-7), Dec 22-24,2014.

[11] M.F. Hyder, M.A. Ismail, H. Ahmed, "Performance Comparison of Hadoop Clusters Configured on Virtual Machines and as a Cloud Service", in proc. of 2014 International Conference on International Technologies,ISBN: 978-1-4799-6088-0, Page No (60-64), Dec 8-9,2014.

[12] Hadoop Tutorial [online]. Available: https:// hadoop.apache.org

AUTHOR'S PROFILE

Ruchi Mittal is a student of M.Tech in Computer Science at the Rayat Bahra Group of Institutes, Patiala(Punjab). She received her B.Tech from PTU. Her research area includes Big Data Analysis using Hadoop and Hadoop in heterogeneous networks.

.