International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

Performance Analysis of Multi-Node Hadoop Clusters using Amazon EC2 Instances

Ruchi Mittal[1], Ruhi Bagga[2]

[1, 2]Rayat Bahra Group of Institutes, Punjab Technical University, Patiala, Punjab, India

Abstract: ***Hadoop, an open source implementation of MapReduce model, is an effective tool for handling, processing and analyzing unstructured data generated these days by different cloud applications. Hadoop considers its nodes to be homogeneous in terms of their processing capability in a cluster. But in real word applications nodes in a cluster are heterogeneous in terms of their processing capability. In such cases, Hadoop does not yields effective performance levels In this paper, we had evaluated and analyzed the performance of WordCount MapReduce application using Hadoop on Amazon EC2 using different Ubuntu instances. The performance has been evaluated both on single node and multi-node clusters. Multi-node clusters include both the homogeneous and the heterogeneous clusters. The performance is evaluated in terms of execution time of the application on different file sizes.***

Keywords: Cloud Computing, Hadoop, MapReduce, Homogeneous & Heterogeneous Hadoop cluster

1. Introduction

Nowadays, humongous amounts of data are generated continuously by the use of various applications which range from Business Computing, Internet, Social Media (e.g. Facebook, Twitter) to Scientific Research. With the growing size of this data every day the need to handle, manage and analyze the same is also growing. To handle such large volume of unstructured data, MapReduce has proven to be an efficient technique. MapReduce, first proposed by Google in 2004, is an efficient programmable framework for handling large data in a parallel, distributed manner in a cluster of many systems.

Hadoop, part of the Apache project sponsored by the Apache Software Foundation, is an open source implementation of the MapReduce programming paradigm that allows distributed processing of large datasets on programmable clusters of computers. Hadoop is built to handle terabytes and petabytes of unstructured data. It is used to handle large datasets over extensive applications. Hadoop is the answer to many questions generated by the challenges of Big Data.

Heterogeneity and Data Locality are the main factors affecting the performance of the Hadoop system because of its architecture. In the classic homogeneous Hadoop system, all the nodes have the same processing ability and hard disk capacity. However in the real world applications the nodes may be of different processing ability and hard disk capacities. With the default Hadoop strategy, the faster nodes may finish processing their local data at a greater speed. After finishing the task with the local data faster nodes can then work with non-local data which would be present on slower nodes. This requires more data movement between the nodes, thus affecting the performance of Hadoop.

In this work, we have done performance analysis of Hadoop in a Single Node Cluster, that is, in a pseudo-distributed mode on different physical machines by executing a word counting application on different input data sizes. For performance evaluation of Hadoop in Multi Node Clusters we have used Amazon EC2 Ubuntu instances, service provided by Amazon Web Service (AWS). We had evaluated the performance analysis of Hadoop on both homogeneous and heterogeneous multi-node clusters and performance comparison has been made of both describing the experimental results.

2. 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, running parallel computations on data 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.

In figure 1, the client machine has Hadoop installed with all the cluster settings. It is neither a NameNode nor DataNode. The role of the client machine is to load the data into the cluster, submitting the MapReduce jobs to the nodes and then monitoring the job processing and retrieving the results when job is finished. Hadoop runs best on the Linux machines, working directly with the underlying hardware.

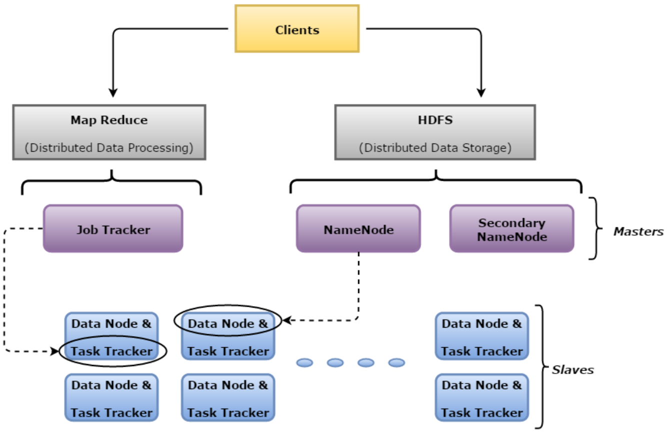

Figure 1: Hadoop Architecture

2.1 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. DataNodes are the slave nodes which perform the actual working like block operations. There is also Secondary NameNode in HDFS which acts like the housekeeping node of NameNode.

The Client partitions the data into blocks which are then stored on the DataNodes in a cluster. 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 a different node in a different rack. This block replication is for prevention of data loss in case of DataNode failure. 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 holds all the metadata of the whole file system in a file named fsimage and oversees the health of all the DataNodes and coordinates access to data in a cluster. The NameNode is the central controller of HDFS. It maps the whole data of the cluster to different DataNodes. It also keeps track of all the transactions carried out in the cluster. When the NameNode is down, the whole cluster is down.
- DataNodes are the slave nodes of the HDFS which performs the actual operations on the requests submitted by the client to the NameNode. DataNodes communicate with the NameNode by sending heartbeat messages every three seconds via TCP handshake. Every tenth heartbeat is a Block Report where the DataNodes tell NameNode about all the data blocks it has using which NameNode builds its metadata.

  Secondary NameNode acts like the housekeeping node of the NameNode. It checks the file system for changes periodically and merges 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.

2.2 MapReduce

MapReduce is the programmable framework for processing 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 exploiting master/slave architecture.

In this, processing is carried out in two different phases namely Map Phase and Reduce Phase. In map phase, the input is partitioned 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 shuffles, sorts and combines all the outputs to produce a single output. The role of JobTracker and TaskTracker is described as follow:

- 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 TaskTracker which is still alive.
- TaskTracker receives the job from the JobTracker and breaks them into map and reduce tasks. It executes the tasks and reports the status update to the JobTracker by sending heartbeat messages and sends the output at the end.

3.3 Amazon EC2

AWS, a cloud-computing platform, is a collection of remote computing services offered by Amazon.com. Amazon Elastic Compute Cloud (EC2) is a well known web service provided by the AWS which provides resizable compute capacity in the cloud much faster and cheaper than building physical servers. With Amazon EC2, one can easily create, launch, reboot and terminate the virtual servers. It allows the user to configure these virtual servers according to their requirements. Amazon, Google, Microsoft, IBM, Apple, Citrix, VMware, etc. are some of the major cloud service providers as of today.

3. Experimental Setup

The experiments were setup using Amazon Elastic Cloud Computing (EC2) platform by Amazon Web Service cloud. For our series of experiments, we have used different types of EC2 instances which include different t2 and m3 instances. The hardware configuration of all the instances which we have used for our work is shown in table 1.

Place table titles above the tables.

Table 1: Hardware configuration of selected EC2 instances

Instance Type	*vCPU*	*Memory (GiB)*	*Clock Speed (GHz)*
t2.micro	1	1	upto 3.3
t2.medium	2	4	upto 3.3
t2.large	2	8	upto 3.0
m3.medium	1	3.75	2.5
m3.large	2	7.5	2.5

The Amazon EC2 t2 instances are based on Intel(R) Xeon(R) processor and m3 instances are based on Intel(R) Xeon E5-2670 v2 processors. The operating system on these instances is 64-bit Ubuntu server 12.04.3 LTS. All instances are populated with 24 GB SSD memory. Hadoop installed version is 1.2.1 and 64-bit Java OpenJDK 1.7.0_25 is installed on all EC2 instances. We have used EC2 t2.micro type nodes for homogeneous cluster. For heterogeneous cluster, different mentioned nodes have been selected. The MapReduce application used for performance evaluation is "wordcount", which count the number of occurrences of each word in a given input file.

4. Results and Observations

In this section, we have compared the results of MapReduce application "wordcount" executed on different file sizes ranging from 300MB to 6GB. This application is executed on different clusters configured with 6, 8 and 10 number of nodes with one node as a NameNode and one Secondary NameNode and all other as DataNodes. The clusters are taken to be homogeneous as well as heterogeneous in their configuration.

4.1 Results of Homogeneous Clusters

The "wordcount" application is executed on three different homogeneous clusters with 6 nodes, 8 nodes and 10 nodes on different input file sizes containing 300MB to 6GB of text data to count the number of occurrences of each word in the files. . All these multi node clusters are populated with t2.micro EC2 instances. All these clusters have one NameNode, one Secondary NameNode and other nodes as DataNodes. Table 2 and Figure 2 shows the results and performance comparison of these three homogeneous clusters.

Table 1 : Results of "wordcount" on Homogeneous Clusters

Input Data Size (GB)	Execution Time (in seconds)		
	6 Node Cluster (4 Slaves)	*8 Node Cluster (6 Slaves)*	*10 Node Cluster (8 Slaves)*
0.28	75	54	53
1.08	161	148	158
1.90	272	221	209
2.70	403	299	282
3.52	503	384	324
4.34	595	447	412
5.16	697	538	538
6.00	805	612	557

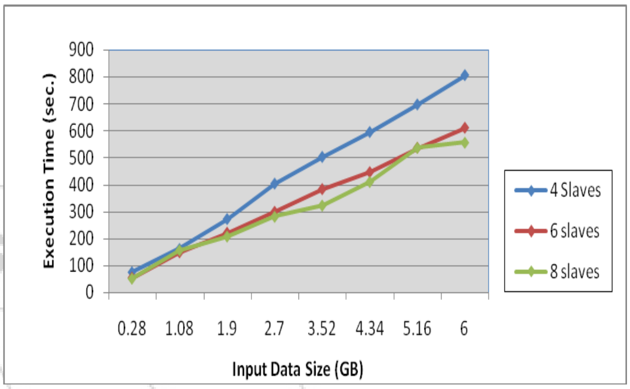

Figure 1: Performance comparison of Homogeneous Clusters

4.2 Results of Heterogeneous Clusters and its comparison with Homogeneous Clusters

We have configured three heterogeneous clusters having 6 number of nodes i.e. one NameNode, one Secondary NameNode and four DataNodes. These three clusters differ in terms of their configuration i.e. the type of DataNodes selected for the cluster to test different levels of heterogeneity. The cluster configuration of all these three clusters has been shown in the table 3.

Table 2: Configuration of Heterogeneous Clusters

Heterogeneous Cluster	**NameNode & Secondary NameNode**	**DataNodes (4 Slaves)**
Cluster 1	t2.micro	t2.micro, t2.micro, t2.micro, t2.large
Cluster 2	t2.micro	t2.micro, t2.medium, m3.medium, m3.large
Cluster 3	t2.micro	t2.micro, t2.medium, t2.large, m3.large

Table 4 shows the execution time taken by different heterogeneous cluster to execute the same "wordcount" application on the same input text files as in the homogeneous clusters. The graph in figure 3 depicts the performance comparison of these clusters with each other and also shows the comparison with homogeneous clusters.

Table 3 : Results of "wordcount" on Heterogeneous Clusters

Input File Size (GB)	Execution Time (in seconds) Heterogeneous Cluster		
	Cluster 1	*Cluster 2*	*Cluster 3*
0.28	79	80	74
1.08	143	191	132
1.9	253	278	232
2.7	337	369	305
3.52	445	490	337
4.34	524	543	463
5.16	596	677	541
6	670	786	625

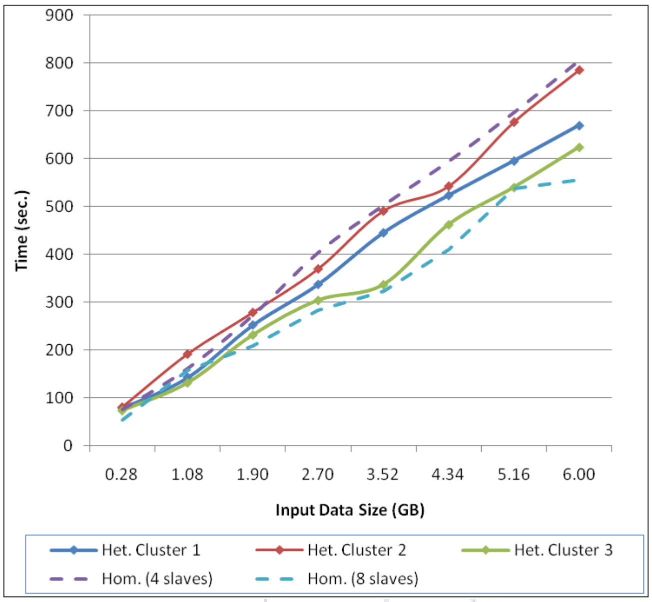

Figure 3: Performance comparison of Heterogeneous Clusters with Homogeneous Clusters

5. Conclusion

In this paper, we have analyzed and observed the results of MapReduce application "wordcount" on different homogeneous and heterogeneous cloud based Hadoop clusters on different input files sizes. In homogeneous 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.

For heterogeneous clusters, it has been concluded that with right combination of heterogeneous nodes, the Hadoop cluster performance can be made better than that of homogeneous one. But a wrong combination can result in performance degradation and additional overhead between the DataNodes.

6. Future Scope

In real world applications, heterogeneous nodes are unavoidable. Our future work aims in the direction of developing some methods which can suggest some combinations for heterogeneous cluster for enhancing their performance as compared to the present scenario.

References

[1] Matei Zaharia, Andy Konwinski, Anthony D. Joseph, Randy Kartz, Ion Stocia, "Improving MapReduce Performance in Heterogeneous Environments", In In Proceedings of 8th USENIX Symposium on Operating Systems Design and Implementation, 29-42, ACM Press, 2008.

[2] Jiong Xie, Shu Yin, Xiaojun Ruan, Zhiyang Ding, Yun Tian, James Majors, Adam Manzanares, Xiao Qin, "Improving MapReduce Performance through Data Placement in Heterogeneous Hadoop Clusters",In Proceedings of International Symposium on Parallel and Distributed Processing, Workshops and PhD Forum,1-9 Apr. 2010.

[3] B. Thirumala, N.V. Sridevi, V. Krinshna Reddy, L.S.S. Reddy, "Performance Issues of Heterogeneous Hadoop Clusters in Cloud COmputing", Global Journal of Computer Science and Technology, v. 11, 2011.

[4] Visalakshi P and Karthik TU, "MapReduce Scheduler Using Classifiers for Heterogeneous Workloads",in IJCSNS, 2011, v. 11, 68-73.

[5] Yuanquan Fan, Weiguo Wu, Haijun Cao, Huo Zhu, Xu Zhao, Wei Wei, "A Heterogeneity Aware Data Distribution and Rebalance Method in Hadoop Cluster", In Proceedings of 7th ChinaGrid Annual Conference, 255-264, IEEE 2012

[6] Xuelian Lin, Zide Meng, Chuan Xu, Meng Wang,"A Pratical Performance Model for Hadoop MapReduce", In Proceedings of the International Conference on Cluster Computing Workshops, 231-239, IEEE 2012.

[7] M. Maurya, S. Mahajan,"Performance Analysis of MapReduce Programs on Hadoop Cluster", In Proceedings of World Congress on Information and Communication Technologies, 505-510, 2012.

[8] M. Ishii, Jungkyu Han, H. Mankino,"Design and Performance Evaluation for Hadoop Clusters on Virtualized Environment",In Proceedings of International Conference on Information Networking, 244-249, 2013.

[9] Zhuoyao Zhang, Ludmila Cherksova, Boon Thau Loo,"Performance Modeling od MapReduce Jobs in Heterogeneous Cloud Environments", In Proceedings of the 6th International Conference on Cloud Computing, 839-846, IEEE 2013.

[10] Wei Dei, Mostafa Bassiouni, "An Improved Task Assignment Scheme for Hadoop running in Clouds", Journal of Cloud Computing, v. 2:23,1-16,2013.

[11] J. Nandimath, E. Banerjee, A.Patil, P. Kakade, "Big Data Analysis using Apache Hadoop", In Proceedings of 14th International Conference on Information Reuse and Intergration, 700-703, IEEE 2013.

[12] Parth Gohil, Dweepna Garg, Bakul Panchal, "A Performance Analysis of MapReduce Applications on Big Data in Cloud based Hadoop", International Conference on Information Communication and Embedded Systems (ICICES), 1-6, 2014

[13] M.F. Hyder, M.A. Ismail, H. Ahmed, "Performance Comparison of Hadoop Clusters Configured on Virtual Machines and as a Cloud Service", In Proceedings of

International Conference on International Technologies, 60-64, 2014.

[14] Ivanilton Polato, Reginaldo Re, Alfredo Goldman, Fabio Kon, "A Comprehensive view of Hadoop Research- A Systematic Literature Review", Elsevier- Journal of Network and Computer Applications,v. 46, pp. 1-25, 2014.

[15] Javier Conejero, Blanca Caminero, Carmen Carrion, "Analysing Hadoop Performancein a Multi-user IaaS Cloud", In Proceedings of International Conference on High Performance Computing & Simulation (HPCS), 399-406, 2014.

[16] Jessica Hartog, Renan DelValle, Madhusudhan Govindaraju, Maichael J. Lewis, "Configuring A MapReduce Framework For Performance Heterogeneous Clusters", In Proceedings of International Congress on Big Data, 120-127, IEEE 2014.

[17] Amazon EC2. https://console.aws.amazon.com

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

[19] T. White, Hadoop: The Definitive Guide, O'Reilly Media, Yahoo! Press, 2009.