

# Performance Analysis of Hadoop Map Reduce on Eucalyptus Private Cloud

Jobby P Jacob

Dept of Computer Science & Engg.- R&D Centre  
East Point College of Engineering & Technology,  
Bangalore, India

Anirban Basu

Dept of Computer Science & Engg.- R&D Centre  
East Point College of Engineering & Technology,  
Bangalore, India

## ABSTRACT

The cost effectiveness and the ease of maintenance are the reasons behind the increasing popularity of Cloud Computing. The need to reduce the execution time of programs on Cloud platforms have led to development of Hadoop [12]. This paper analyzes the performance of K-Means Clustering Algorithm when running on Hadoop MapReduce on Eucalyptus [5] platform. Running Eucalyptus for Hadoop requires lot of customization for the software to run as discussed here. Several tools like Ganglia, TestDSFIO.java, Linux performance measuring tools have been used to measure the performance. The paper discusses how performance of K-Means Clustering Algorithm scales up with number of nodes on Eucalyptus cloud. Results of measurement of the disk, network, memory bandwidth, data throughput and average I/O are presented here.

## Keywords

BigData, Hadoop, MapReduce, Eucalyptus Cloud, Ganglia.

## 1. INTRODUCTION

Hadoop MapReduce had grown in popularity over the recent years due its robustness, ease and effectiveness when processing large amounts of data. This paper evaluates the performance of K-Means Clustering algorithm implemented using Hadoop MapReduce framework on a Eucalyptus private cloud. For the evaluation of performance, work involved in running K-Means clustering algorithm on Hadoop cluster on cloud and extended the Map Reduce to run on multiple nodes. Measurements were carried out and results indicate that Hadoop MapReduce can be run efficiently on Eucalyptus private cloud for this algorithm and is easily scalable.

## 2. EARLIER WORK

In [2], Chen He, Derek Weitzel, David Swanson, Ying Lu discussed experiments by running Hadoop Map Reduce on a grid which provides a free, elastic, and dynamic MapReduce environment on the opportunistic resources of the grid. For Hadoop evaluation, they successfully extended Hadoop to 1100 nodes on Grid. From the evaluation, Chen and his team found that the unreliability of the grid makes Hadoop on the grid very challenging. K-means Clustering Using Hadoop Map Reduce by Grace Nila Ramamoorthy[1] evaluated the performance of K-Means clustering using Hadoop Map Reduce on standalone servers infrastructure. This model works well, but when it comes to scalability and efficient use of resources and need for rebuilding environment for different projects it becomes complex and very time consuming. It is also not a good model when scalability is a requirement as individual nodes have to be setup to scale up. Grace also noted that the number of nodes available for map tasks affect performance and more the number of nodes, the better is the

performance and MapReduce a valuable tool for clustering larger datasets that are distributed and cannot be stored on a single node.

Clustering in the cloud by Xuan Wang[3] chose to run clustering techniques on Hadoop MapReduce Framework on the cloud. For the cloud infrastructure, he used the public cloud for this purpose. And always public clouds work on the internet and when it comes to transferring large data, it's a problem. However he has determined that clustering technologies can be run on public clouds.

## 3. APACHE HADOOP

Apache Hadoop is a software written in Java which brings to the table an open source platform that enables data centric application to run parallel in a distributed environment. This framework has proved to be an effective way to run an application in parallel especially dealing with terabytes of data. Hadoop enables applications to work with thousands of nodes and terabytes of data, without concerning the user with too much detail on the allocation and distribution of data and calculation. Hadoop is open source and distributed under Apache license. The main components of Hadoop are: MapReduce and HDFS.

### 3.1 MAPREDUCE

MapReduce[5] is a programming model that works on large datasets. Many organizations use MapReduce model for computing when they have huge datasets and need to process them within short time. MapReduce works by breaking the processing into two phases: the map phase and the reduce phase[10]. Each phase has key-value pairs as input and output, the types of which may be chosen by the programmer. In MapReduce, the Map function processes the input in the form of key/value pairs and generate intermediate key/value pairs, and the Reduce function process all intermediate values associated with the same intermediate key generated by the Map function[10]. Reduce program have to written in such a way as it can accept the key value pairs of map and then process it.

MapReduce performs series of operation on the input key and value. As many companies now adopt Hadoop, the requirements of what input varies from flat files to databases. Input is split of chunks. In case of database, one can take an example of rows from the table. Split actually doesn't contain the data. As always it is just a reference After splitting, the split component is sent to jobtracker and map tasks are scheduled to process them.

Map functions are evenly distributed between all nodes in cluster and locality of the data can be considered unknown. There are also better ways to balance the data and jobs that are local to the data. The map phase receives a key value pair that

can be a generic type and outputs zero or more key value pairs that may be of some other types also. The input type and output type can be same or it can be different. The reduce function process the results that are the output from map phase for every key that is unique. Reducer function receives values and outputs zero or more key value pairs. MapReduce programs interact with framework for processing large tasks. The framework takes care of assigning task to map and reduce phase, rerunning of tasks that are failed, splitting input to feed map phase, giving map output to reducer and receiving output from reducer.

### 3.2 HDFS

Hadoop distributed file system is designed for containing very large files that are terabytes in size. It is built around the concept of write once and read many times and generally in case of high compute application this theory is valid and this is the reason why HDFS adds to the overall performance of the file system and MapReduce as a whole. HDFS file system and the data files are distributed and stored in chunks. And HDFS file system property can be attributed to data being local to the node and rather avoid the costly data move between nodes in cluster.

It can keep multiple copies of data which increases reliability. One can have an option to run file system integrity check which can correct file system inconsistencies, missing blocks etc. This feature is plus in tandem with the commercial file systems available now.

HDFS in a normal enterprise deployment consists of a single NameNode and multiple datanodes. Namenode is considered as HDFS master and all datanodes works as slave. In order to ensure availability of NameNode, redundant files are stored in DataNodes. Filesystem's metadata is stored in NameNode. Metadata contains information about file system permissions, last access time etc. When a user wants to read a file it uses a Hadoop HDFS client that contacts the NameNode. The NameNode then fetches the block locations and return the locations to the client, forcing the client to do the reading and merging of blocks from the DataNodes.

NameNode provides interface to the users who can access some useful information from the user interface provided through http. Data like file system capacity, name node logs, its configured capacity, how many nodes HDFS cluster is made of, etc.

## 4. EUCALYPTUS CLOUD

Cloud computing[6] can be termed as a system that contains different modules to integrate fundamental components of an enterprise computing to function or control as a single point of management. It provides access to large pools of data storage, network bandwidth, system resources that can be distributed to number of machines that act as a standalone machine.

Eucalyptus is an framework developed using the open-source standards and is referred as Infrastructure as a Service (IaaS)[10], systems which give IT architects to control a large IT environment using a single software irrespective of end users demand. Apart from this, end user gets an ability to easily build and deploy systems, controlled and scaled up with much ease. Different operating systems can be deployed on Eucalyptus cloud. This provides the flexibility to ensure end users need when they deploy their applications. Eucalyptus software is installed on top of CentOS termed as Host operating system and the clients can be termed as virtual machines. Eucalyptus software is configured during the initial

installation with network information, storage arrays, how IP should be assigned and what kind of data store to use. Eucalyptus then contacts different daemons running for different components to determine the setup and layout of the systems. With this information Eucalyptus manages the dynamic creation, distribution and sharing of resources to multiple virtual machines spread across the physical hardware which are part of the Eucalyptus system. The main components of Eucalyptus are:

### 4.1 Walrus

Walrus[11] is storage sub system where data is stored similar to Amazon S3[13]. Data is logically grouped into buckets and have the same API to read and write data in the bucket.

### 4.2 Cloud Controller (CLC)

The Cloud Controller [11] is the component that controls the virtual machines operation like starting, stopping, assigning of systems resources like RAM and CPU, and polling of the components frequently to determine the status of each system resources assigned. It manages the user access database and keys required for administration. It's also an entry for the end users and administrators to configure and use the system.

### 4.3 Storage Controller

This is the equivalent to the EBS found in Amazon[13]. SAN storage, Local disk arrays, local disks, NFS can be added to the storage controller nodes. Depending on the storage assigned, we can group the storage needs depending on what kind of performance is expected from file system. Storage can be dynamically assigned.

## 5. PERFORMANCE ANALYSIS OF HADOOP

Experiments were conducted by running a K-Means clustering job with three iterations to cluster junk of emails. Hence measured the performance using monitoring tool Ganglia to capture CPU, memory, disk I/O, network transfer rate during the time when job was run.

K-Means clustering performance:

The K-Means clustering program was run with chunk of emails and used to analyze how large iterative jobs perform on Hadoop when run on Eucalyptus. The overall results showed excellent performance.

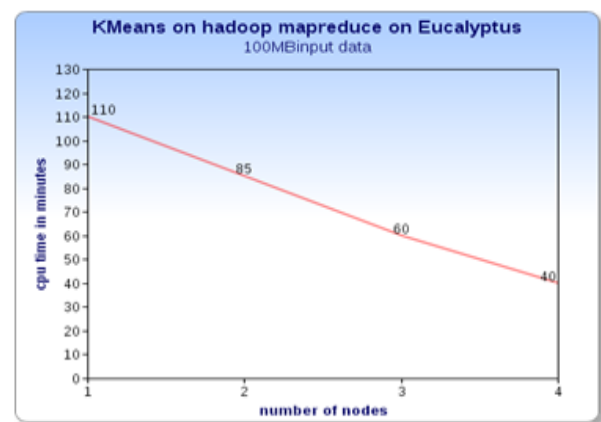
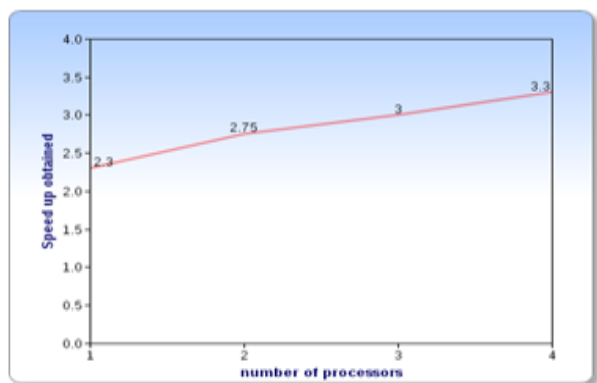


Figure 1

Figure 1 shows the variation of performance of K-Means clustering performance by increasing number of nodes. From

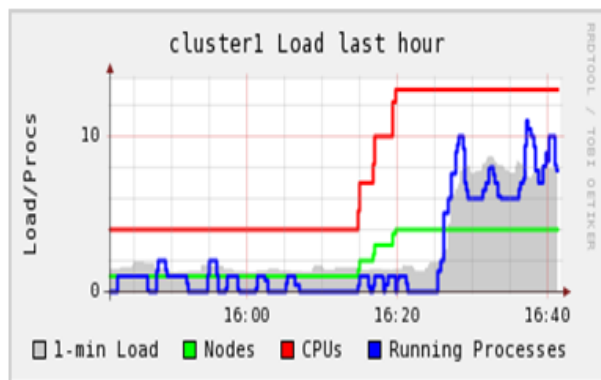
the graph it can be identified that the execution time decreases as number of nodes increases.



**Figure 2**

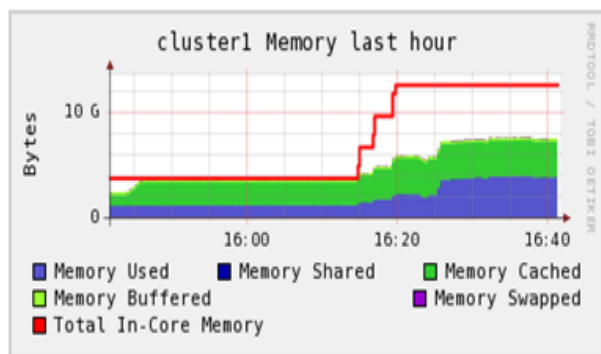
Figure 2 shows the Speed up obtained by varying number of nodes.

The CPU utilization is considerably higher but not high enough to affect the node controllers of Eucalyptus. Since some CPU and memory is allocated to run the cloud environment, high CPU or Memory usage by MapReduce programs are not throttling the cloud environment.



**Figure 3**

Figure 3 shows the entire Cluster-Load average when K-Means job was run with 3 iterations. This is measured right after the job started running. The graph measures the load average of the program.



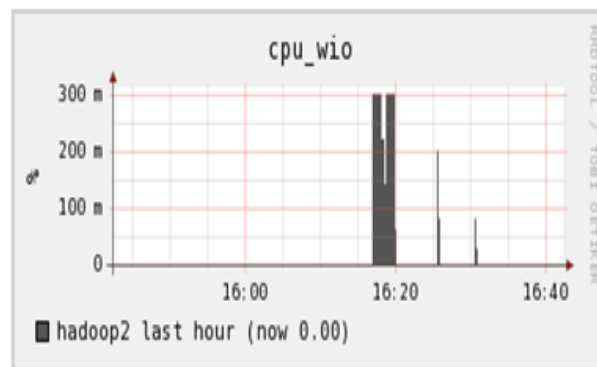
**Figure 4**

Figure 4 shows Entire cluster Memory Statistics when K-Means job was run with 3 iterations. The graph shows how

much memory compared to total memory was being utilized by the running process.

Using three VMs clearly improves execution speed and four VMs also improves the map/shuffle performance, but with a decrease in CPU utilization.

Due to the limitations of the testing environment, I/O's were written to the local disk using iSCSI with 100Mbps network. Although there are limitations in using SAN disks or other high end storage devices, local disk performed well to the extensive reads and writes to the disk. From the monitoring tool TestDFSIO, the process waiting for I/O to complete clearly showed the bottleneck.



**Figure 5**

Figure 5 shows Hadoop datanode : CPU wait time for I/O Process to complete. This graph shows hadoop2 wait time for I/O process after job started running.

## 6. CONCLUSION

While Hadoop MapReduce is designed to be used on top of physical commodity hardware servers, testing has shown that running Hadoop MapReduce in private cloud supplied by Eucalyptus is viable and very efficient. The results showed the performance of MapReduce job running in four nodes is good. It was noted that the increase in number of nodes boosted performance significantly. And also the performance gain increases as number of nodes scale up.

The need for building individual standalone machines for different user requirements have become things of the past. Creating or running an OS is much faster as pre-build images can be launched very easily and customized according to user requirements. Cloud also gives the user the ability to supply more machines when needed, as long as it is not reaching the physical upper limits of the underlying host machines.

Eucalyptus private cloud should be improved to accommodate different user needs when running an image. This needs to be improved so that effort to bring up instance initially will not take much time. Hadoop design of working on HDFS file systems still needs much improvement. Hadoop can look into integrating GFS[5]file system.

## 7. REFERENCES

- [1] Grace Nila Ramamoorthy:K-Means Clustering Using Hadoop MapReduce. Published by UCD School of Computer Science and Informatics.
- [2] Chen He, Derek Weitzel, David Swanson, Ying Lu. HOG: Distributed Hadoop MapReduce on the Grid Published by 2012 SC Companion: High Performance Computing, Networking Storage and Analysis.

- [3] Xuan Wang, "Clustering in the cloud: Clustering Algorithms to Hadoop Map/Reduce Framework" (2010), Published by Technical Reports-Computer Science by Texas State University.
- [4] Apache Software Foundation, Hdfs user guide <http://hadoop.apache.org/hdfs/docs/current/hdfsuserguide>
- [5] MapReduce tutorial Apache Hadoop 1.2.1 documentation by Hadoop wiki.
- [6] Eucalyptus Systems, Inc. Eucalyptus 3.3.1, Eucalyptus Administration Guide (2.0), 2010.
- [7] J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. Proceedings of 6th Symposium on Operating System Design and Implementation, Published by Communications of ACM, Volume 51 Issue 1, January 2008.
- [8] Cloudera. Cloudera's distribution including Apache Hadoop.
- [9] A. T. Velte, T. J. Velte, and R. Elsenpeter. Cloud Computing- A Practical Approach, Published by The McGraw-Hill Companies, 2010.
- [10] Tom White Hadoop- The Definitive Guide, Published by O'Reilly Media/Yahoo Press, 2nd edition, 2010.
- [11] Huan Liu and Dan Orban, Cloud MapReduce: a MapReduce Implementation on top of a Cloud Operating System. Published in Cluster, Cloud and Grid Computing (CCGrid) 2011, 11th IEEE/ACM International Symposium.
- [12] Weizhong Zhao, Huifang Ma, Qing He, Parallel K-Means clustering based on MapReduce Published by The Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences.
- [13] The Apache Hadoop Ecosystem, University of Cloudera, Online Resources.
- [14] Amazon. Amazon elastic block storage (EBS), AWS documentation by Amazon Elastic Compute Cloud User Guide.
- [15] Blaise Barney, Introduction to Parallel Computing, Published by Lawrence Livermore National Laboratory.