2013

Performance Evaluation of Data Intensive Computing In The Cloud

Bhagavathi Kaza
University of North Florida, n00697704@ospreys.unf.edu

PERFORMANCE EVALUATION OF DATA-INTENSIVE COMPUTING IN THE CLOUD

by

Bhagavathi Kaza

A thesis submitted to the
School of Computing
in partial fulfillment of the requirements for the degree of

Master of Science in Computer and Information Sciences

UNIVERSITY OF NORTH FLORIDA
SCHOOL OF COMPUTING

August, 2013

The thesis "Performance Evaluation of Data-Intensive Computing in the Cloud” submitted by Bhagavathi Kaza in partial fulfillment of the requirements for the degree of Master of Science in Computer and Information Sciences has been

Approved by the thesis committee: Date

________________________________ ____________________

Dr. Sanjay P. Ahuja
Thesis Advisor and Committee Chairperson

________________________________ ____________________

Dr. Roger Eggen

________________________________ ____________________

Dr. Zornitza G. Prodanoff

Accepted for the School of Computing:

________________________________ ____________________

Dr. Asai Asaithambi
Director of the School

Accepted for the College of Computing, Engineering, and Construction:

________________________________ ____________________

Dr. Mark A. Tumeo
Dean of the College

Accepted for the University:

________________________________ ____________________

Dr. Len Roberson
Dean of the Graduate School

ACKNOWLEDGEMENT

I would like to take this opportunity to express my gratitude to the people without whose support this thesis would not have been possible. I would like to thank my thesis advisor, Dr. Sanjay P. Ahuja for his valuable guidance and support, and I would like to thank my committee members, Dr. Roger Eggen and Dr. Zornitza Prodanoff, for their valuable suggestions. I would also like to thank Dr. Asai Asaithambi and Dr. Karthikeyan Umapathy for attending my thesis defense and providing suggestions. I would also like to thank Mr. Jim Littleton for providing valuable editorial suggestions on my thesis write up. Finally, I would like to thank my husband and family for their continuous support, encouragement and love during the long process in achieving this important goal.

CONTENTS

FIGURES

TABLES

ABSTRACT

Big data is a topic of active research in the cloud community. With increasing demand for data storage in the cloud, study of data-intensive applications is becoming a primary focus. Data-intensive applications involve high CPU usage for processing large volumes of data on the scale of terabytes or petabytes. While some research exists for the performance effect of data intensive applications in the cloud, none of the research compares the Amazon Elastic Compute Cloud (Amazon EC2) and Google Compute Engine (GCE) clouds using multiple benchmarks. This study performs extensive research on the Amazon EC2 and GCE clouds using the TeraSort, MalStone and CreditStone benchmarks on Hadoop and Sector data layers. Data collected for the Amazon EC2 and GCE clouds measure performance as the number of nodes is varied. This study shows that GCE is more efficient for data-intensive applications compared to Amazon EC2.

Chapter 1

INTRODUCTION

The National Institute of Standards and Technology (NIST) defines Cloud Computing as

> A model for enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction [NIST11].

A cloud offers the hardware and software necessary to support an application while providing storage, performance, security and maintenance. Clouds are classified by the deployment and service models they utilize.

The concept of deployment models leads to the classification of Public clouds, Private clouds and Hybrid clouds. Public clouds offer storage and other resources on a pay-per-use basis. Private clouds offer the needed infrastructure through either the internal organization or by third party vendors; however, private cloud pose certain risks relating to scalability, maintenance, and investments. Hybrid clouds, a combination of Public and Private clouds serve the benefits of multiple deployment models and degrees of fault tolerance. Amazon Elastic Compute Cloud (Amazon EC2), Google Compute Engine (GCE), and Microsoft Azure are examples of public clouds, and OpenStack, CloudStack and VMware’s vCloud are examples of private clouds, and CliQr is an example of hybrid cloud.

Cloud service models include Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). IaaS provides a virtual computer platform-capable of running various operating systems through the use of virtual machine images, and provides an infrastructure that can include file -storage, firewall, IP addresses, load balancers etc. With IaaS, users pay for only the resources utilized. Amazon EC2, GCE, Rackspace and Microsoft Azure service platforms are examples of IaaS. PaaS provides a complete computer platform including a choice of operating system, programming environment, database and servers, and the platform's resources can scale automatically to handle the demands of the application. With PaaS, users pay for only the time utilized on the platform. Force.com, AWS Elastic Beanstalk and Google App Engine are examples of PaaS. SaaS provides an on-demand programming environment and database, which allows users to run their applications without the overhead of providing IT support and maintenance. With SaaS, users pay on a per-use model, which typically is in the form of a monthly or annual subscription. Google App Engine and Salesforce.com are examples of SaaS.

1.1 Cloud Architectures Compared

1.1.1 Amazon Elastic Compute Cloud (Amazon EC2)

Amazon EC2 is an IaaS cloud service that provides a resizable computing capacity. Amazon EC2 provides a flexible, web-based interface that allows users to configure the environment and launch instances of their chosen operating system – called Amazon

Machine Images (AMI), as well as select the appropriate application type. Amazon EC2 supports a variety of Linux operating systems including Red Hat, Debian, Ubuntu, and Fedora. Amazon EC2 also supports several instance types including standard, micro, high CPU, high memory, cluster GPU, high storage, cluster compute, high memory cluster and high I/O instances. Each instance type varies in terms of memory capacity, available virtual cores, storage capacity and I/O performance. Amazon EC2 defines the minimum processing unit, referred to as EC2 Compute Unit (ECU), which is the equivalent CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor [AWS13].

In addition to the available instances, Amazon EC2 also provides On Demand Instances, Spot Instances and Reserved Instances based on the purchasing models. On Demand instances allows paying fixed rate per hour without commitment. Spot instances allow paying a one-time low fee and receiving a discount on hourly charge for the instance. Spot instances allows to bid as per the user price for the instance capacity. Users choose the best instance type from the wide variety presented above based on their application needs.

1.1.2 Google Compute Engine (GCE)

Google Compute Engine (GCE) is an open source IaaS cloud service. GCE is a suitable alternative to the Amazon EC2 cloud service. GCE provides a RESTful API for managing cloud resources such as file storage, networking, virtual machine images and launching operating system instances. An access token, provided by OAuth 2.0, is required to authenticate access using the GCE API. GCE supports several instance types including standard, high memory and high CPU. The standard instance type supports most applications that require fewer memory and CPU resources. The high memory instance type supports applications that require greater memory resources, and the high CPU instance type supports applications that require greater CPU resources – the number of virtual cores used by the application. GCE defines the minimum processing unit, referred to as Google Compute Engine Unit (GCEU), which is the equivalent CPU capacity of a 1.0-1.2 GHz 2007 Opteron processor. GCE uses 2.75 GCEU's to represent the minimum processing power of one logical core. Each instance type has two default configurations such as n1-standard-2 and n1-standard-2-d, which differ only in available scratch disk capacity while CPU capacity, memory capacity and persistent disk capacity remains the same. The present research uses High-CPU instances for data intensive applications needing more CPU compared to memory. Choice of the zone is the nearest zone available, which avoids network and latency issues, and the choice of instance type depends on the demands of the user application. Table 1 compares the Amazon EC2 and GCE cloud service features that are significant to the current study.

	Amazon EC2	GCE
Number of cores	2	2
Compute unit	5ECU	5.5GCEU
Processor	Intel Xeon E5-2670	Intel Xeon
Memory	1.7GB	3.60GB
Instance Storage	700GB	10GB root disk + 870GB of additional scratch disk space
Operating System	Ubuntu 12.04	GCEL 12.04
Storage Type	External	Internal

Table 1: Comparison between Amazon EC2 and GCE.

1.2 Big Data

Big data refers to the collection of large, complex data sets, which can be structured or unstructured, and are difficult to process using traditional relational database management tools. Big data refers to large volumes of data which can be terabytes, petabytes or even xetabytes of data. Big data exists across many fields such as scientific research, information technology, astronomy, biological studies and social networks. Examples of big data include text, sensor data, audio, video and log files. Apache Hadoop and Sector are open source frameworks used to process big data to produce useful information.

1.2.1 Apache Hadoop

Apache Hadoop utilizes Master-slave system architecture. Apache Hadoop supports only a single master node, which is responsible for storing and managing the metadata, but it supports multiple slave (worker) nodes that process and store the data. Hadoop uses the Hadoop Distribution File System (HDFS), which is a block-based distributed file system, to distribute an application across the nodes in a cluster. Apache Hadoop ensures fault tolerance to prevent data loss in the event of a system failure by storing the same data on three unrelated nodes, by default; however, the number of nodes used for fault tolerance (referred to as the Replication Factor) is configurable.

1.2.1.1 MapReduce

MapReduce is a programming model used to process large data sets across a distributed collection of nodes in a cluster. MapReduce transforms a list of inputs, assigned to a specific node in the cluster, into a list of outputs using two distinct functions, Map () and Reduce (). The Map () function converts a set of data (input list) into key-value pairs or tuples (output list), and each input list processed by the Map () function produces a different output list. The Reduce () function creates a single output list by reducing the aggregate tuple data from the output lists generated by the Map () function; however, before the Reduce () function is initiated, all of the output lists generated by the Map () function for a node must be exchanged (copied) with all of the output lists generated by all of the other nodes in the cluster through a process called shuffling. MapReduce requires a "driver" method to initialize a job, which defines the locations of the input and output files and controls the MapReduce process. Each node in a MapReduce cluster is unaware of the other nodes in the cluster, and nodes do not communicate with each other except during the shuffling process. Figure 1 illustrates a high-level view of the MapReduce process using three nodes.

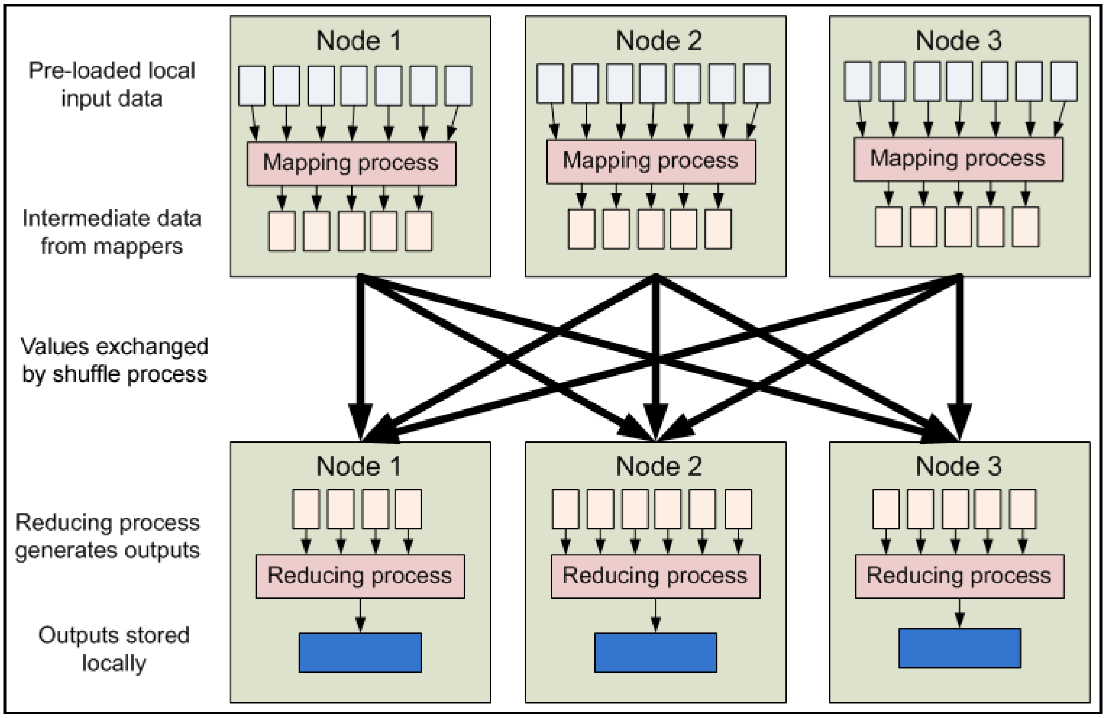

Figure 1: High-level MapReduce Process [Hadoop13].

1.2.2 Sector

Sector is a high-performance, scalable and secure distributed file system capable of supporting large clusters of commodity computers. Sphere is the high-performance parallel data processing engine used to process Sector data files on storage nodes. Features such as high-performance, WAN support, fault tolerance and rule-based data management make Sector a highly optimized solution for data intensive applications. Sector is a valid alternative to Apache Hadoop for data-intensive applications.

Sector utilizes Master-slave system architecture. Sector supports a single master node, which is responsible for storing and managing the metadata, but it supports multiple slave (worker) nodes that process and store the data. Sector ensures fault tolerance to prevent data loss in the event of a system failure by replicating a node's data across a configured number of nodes. Sector uses the UDP- based Data Transfer Protocol (UDT), which is a high-speed data transfer protocol, to transfer data to the nodes in a cluster. Figure 2 depicts the Sector/Sphere architecture.

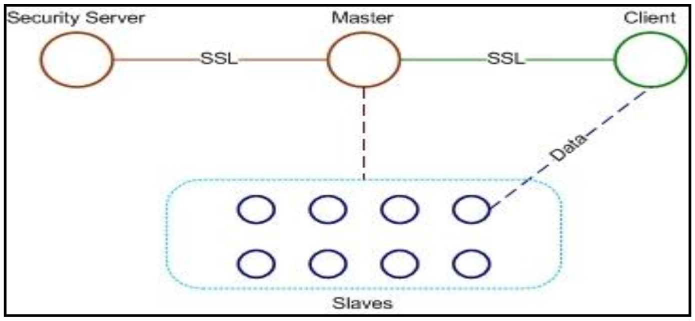

Figure 2: Sector/Sphere Architecture [Gu10A].

Sector divides datasets into one or more files, called Sector Slices, which are stored on each of the slave nodes in a cluster. The client logs onto the Master node through an SSL connection, which transmits the credentials to the Master node. Master sends the credentials to the security server where it checks for the authorization of the client and issues unique session id and file permissions for the client. Sector uses User Datagram Protocol (UDP) for message passing and UDT for data transfer. Sector uses the UDP protocol and Hadoop uses Transmission Control Protocol (TCP). This fact accounts for the fast performance of sector as UDP doesn't need connection setup. Table 2 presents the comparison between Hadoop and Sector.

	Hadoop	Sector
Storage Cloud	Block-based file system	File-based
Programming Model	MapReduce	User Defined Function (UDF) & MapReduce
Protocol	TCP	UDT
Replication	At the time of writing	Periodically

Table 2: Comparison of Apache Hadoop and Sector.

1.3 Benchmarks

1.3.1 TeraSort Benchmark

The TeraSort benchmark, freely available with Apache Hadoop, is widely used to evaluate cloud performance of data-intensive applications [Noll11]. Although the benchmark was designed originally to sort 1TB of data as fast as possible, it can be used to sort any amount of data. The TeraSort benchmark consists of a suite of tools – TeraGen, TeraSort and TeraValidate – used to generate, sort and validate test data, respectively.

TeraGen generates a specified number of 100-byte records for use with the TeraSort benchmark. Each generated record consists of a 10-byte key that is comprised of random characters, a 10-byte row id value and a 78 bytes of filler, which consists of random, alphabetic characters (A to Z) followed by the carriage return and line feed characters (“\r\n”). The generated records are evenly distributed across the available worker nodes in the cluster as input to each node’s Map () function.

TeraSort performs a MapReduce sort of the generated test data, and stores the sorted test data in the specified output directory. TeraSort utilizes a custom partitioner that references a sorted list of N-1 sampled keys that defines the valid key range assigned to each node to constrain the output of its Reduce () function (sample[x–1].key <= key < sample[i].key). This partitioning scheme guarantees the data stored in the output list generated by a node's Reduce () function exists, logically, between the node's logical neighbors. Since fault tolerance is not necessary during benchmark testing, this study uses a replication factor of one rather than the default value.

TeraValidate verifies the data stored in the output files, located in the configured output directory, was sorted correctly by the TeraSort tool, and that each output file appears in the correct logical order. The TeraValidate tool produces output only if it detects issues such as data appearing in the wrong order.

1.3.2 CreditStone Benchmark

The CreditStone benchmark is useful for testing clouds that provide on demand computing capacity. The benchmark generates the data necessary to test the cloud application. The generated data contains synthetic events modeled on credit card transactions with some of the transactions flagged, as well as the computed ratio of flagged to unflagged events.

The data generated by the benchmark contains the following fields:

Transac.ID | Time | Account ID | Merchant ID | Amount | Flag

Transac ID- This field represents the transaction id of the credit card transaction used to track the fraudulent transactions on a card. This field has a length of 11bytes.

Time- This 4-byte field represents the time stamp of the transaction. Data arrives ordered by time stamp.

Account ID- This is a 10-byte field, which represents the account number of the credit card used.

Merchant ID- This 7-byte field represents the unique merchant id.

Amount- This field presents the exact amount on the transaction, which is a valid or fraudulent transaction on user's account.

Flag- This 2-byte field represents a flagged or unflagged transaction on a credit card. A value of 1 indicates a flagged transaction, and a 0 indicates an unflagged transaction.

Data generated by the benchmark is loaded into the HDFS of Hadoop and data layer in Sector, and the Hadoop and Sector settings in the configuration files modified accordingly for the benchmark [Bennett08] [Grossman08].

1.3.3 MalStone Benchmark

MalStone is useful for testing cloud middleware performance for data intensive computing by Open Data Group [Jogalekar00]. MalGen, an open source software package generates site-entity log files. The synthetic data generated follows certain statistical distributions.

Site entity log files consists of events that record a visit to a particular site and is marked as compromised by setting the compromise flag to 1. Uncompromised sites are marked with a 0, and the challenge is to find all the compromised sites in a given log file. Python and Shell scripts, present in the MalGen tool, distribute the large data files across the nodes in the cluster. MalGen uses the power law distribution to model the number of events included in a site. The reason to model the events is that some sites have few entries whereas some other sites have large number of entries [Gu09].

The generated data files consist of 100 byte records one per line. Delimiter is a single byte ASCII value and the data generated consists of four delimiters. The data records generated by MalGen are in the format:

Event ID | Timestamp | site ID | Compromise Flag | Entity ID

Appendix A presents a clear description of each field of a MalStone benchmark record.

1.4 Research Objectives

This study compares the performance of two public cloud services, Amazon EC2 and GCE, running on the IaaS service model. Parameters such as the number of nodes, hardware and software resources and instance types vary while evaluating the performance of each cloud. This study, which extends previous research on benchmarking the Amazon EC2 and GCE clouds to evaluate their performance, evaluates and analyses cloud performance using three benchmarks – TeraSort, MalStone and CreditStone. The literature review presents the previous work carried out on private clouds. It is important to note that no previous work exists on public clouds using the three data-intensive benchmarks selected for this study.

Chapter 2

LITERATURE REVIEW

Several cloud performance studies exist involving the Amazon EC2 and GCE cloud services along Apache Hadoop or Sector and tested using the TeraSort, MalStone or CreditStone benchmarks. These studies measure a cloud's performance while varying parameters such as the number of nodes, hardware and software resources and instance types. Although studies using the TeraSort, MalStone and CreditStone benchmarks exist, no study exists that performs cloud performance tests with all three benchmarks.

2.1 Studies using TeraSort Benchmark

O'Malley *et al* in Yahoo made a record with TeraSort benchmark by running the code generated by TeraGen on a 910-node cluster and sorted 1TB of data in 209 seconds. The cluster consisted of 910 nodes, 4 dual core Xeons at 2.0 GHz per node, 8GB RAM per node, a gigabit Ethernet cable on each node, 40 nodes per rack on Hadoop 0.18 version [Malley08].

A study of Apache Hadoop and Sector using the TeraSort benchmark performed by Gu *et al* concluded that Sector is approximately two times faster than Apache Hadoop [Gu09]. The study, consisting of four 32-node racks located in three geographic regions of the United States (Baltimore, Chicago and San Diego) also indicated that Sector scaled better than Apache Hadoop as the number of nodes increased. Another study performed using the TeraSort benchmark run on 118 nodes located in the same region, required 1526sec with Sector and 3702 sec with Hadoop indicating Sector's better performance compared to Hadoop.

Grossman *et al* in [Grossman09A], compare Hadoop and Sector on Teraflow Testbed. The testbed constitutes a small cluster of six nodes with 10GB/s network connection on dual-core Opteron servers. Each Opteron has 2.4GHz CPU and 4GB memory. The experimental results indicate the response time for 10 GB data file. Hadoop processes the data file on single node in 1708 sec while Sector took only 510 sec.

2.2 Studies using CreditStone Benchmark

A study of Apache Hadoop and Sector using the CreditStone benchmark performed by Gu et al. concluded that Sector performed faster than Apache Hadoop [Gu09]. The study consisted of 89 nodes located in three geographic regions of the United States (Baltimore, Chicago and San Diego), which processed a 2.5TB dataset, and the drive capacity was increased as the study progressed.

A study of Apache Hadoop and Sector using the CreditStone benchmark performed by Grossman *et al.* concluded that Sector scored better response times than Apache Hadoop [Grossman09B]. The study consisted of up to 117 nodes located in up to four locations in three geographic regions of the United States (Baltimore, Chicago and San Diego), and the number of nodes and locations were increased as the study progressed.

2.3 Studies using MalStone Benchmark

A study of Apache Hadoop and Sector using the MalStone benchmark performed by Bennett *et al* concluded that Sector was 2.5 times faster than Apache Hadoop [Bennett10]. The study consisted of up to 20 nodes and a dataset containing 10 billion records.

A study of Apache Hadoop and Sector using the MalStone benchmark performed by Gu *et al* concluded that Sector scored better response times than Apache Hadoop [Gu09]. The study consisted of four racks with up to 120 nodes and a dataset of 1TB, and the number of nodes increased as the study progressed.

The MalStone benchmark, tested on a DataRush private testbed, using 1TB log file on a machine of four AMD 8435 processors running at 2.6GHz with 24 cores, 64GB memory, and using RAID system with 5 SATA drives for I/O optimization, resulted in Sector's response time of 68min and 840 min on Hadoop [Pervasive10].

Chapter 3

RESEARCH METHODOLOGY

This study evaluates the performance of the Amazon EC2 and GCE cloud services as platforms for data-intensive computation. After installing the latest versions of Apache Hadoop and Sector on both cloud environments, as well as the data intensive benchmarks TeraSort, MalStone and CreditStone, testing of the clouds proceeds as the number of nodes varies from one to eight and the metrics are measured.

The study uses the TeraSort benchmark on Apache Hadoop and Sector on the Amazon EC2 cloud service with varying dataset sizes – 1GB, 10GB, 100GB and 1TB (100GB and 1TB are data-intensive), and other metrics such as the number of nodes, data size varied while measuring response times. The study also uses the MalStone and CreditStone benchmarks in the same manner as the TeraSort benchmark. The study also performs the TeraSort, MalStone and CreditStone benchmarks on the GCE cloud service in the same manner as performed on the Amazon EC2 cloud service.

The study conducts tests with each benchmark five times, and computes the average response time. Conclusions, based on the statistical analysis of the test results were made on the performance of the Amazon EC2 cloud service using Apache Hadoop and Sector as well as on the GCE cloud service.

Chapter 4

TESTBED SETUP

4.1 Creating Clusters On the Amazon EC2 Cloud Service

StarCluster, an open source cluster-computing toolkit designed to automate cluster creation and configuration, creates the cluster on the Amazon EC2 cloud service. StarCluster configures security groups with various firewall rules on each cluster and names each node appropriately (master node: master, slave nodes: node-001, node-002, etc.), and creates clusters on the cloud with their own Amazon Elastic Block Store (EBS) volume [StarCluster13]. A detailed explanation of creating a cluster on the Amazon EC2 cloud service using the StarCluster toolkit is available in Appendix B.

4.2 Creating Clusters On the Google Compute Engine Cloud Service

The Google Compute Engine launches a cluster of eight nodes to test the benchmarks on each node and compare the performance for EC2 and GCE clouds. The installation process on GCE requires some prerequisite software such as maven, git and the Java JDK. Use the appropriate commands to install the prerequisite software based on the host operating system used.

Commands used for installation on Centos are:

```
$ yum install maven
$ yum install git
$ yum install jdk
```

The head node or the coordinator node is responsible for sharing the information among different nodes and also to pass data from the cloud to the storage and back from the storage to the cloud. Num_slaves is the number of nodes that the user desires. Cluster is set to be up and running once the master nodes on Hadoop and Sector are up. The ./tools/status.py script manually polls slave nodes to determine their status. Instances run their own REST agent, snitch, which the coordinator polls to know when the cluster is ready.

To add additional nodes to the cluster:

```
$./tools/add_slaves.py num_slaves
```

To destroy the cluster after the job is completed (prevents the cluster from accruing unwanted charges on the cloud service):

```
$./tools/teardown.py
```

A detailed explanation of creating a cluster on the GCE cloud service is available in Appendix C.

4.3 Hadoop setup

Apache Hadoop supports the Linux and Windows operating system; however, it fully supports only the Linux operating system. Apache Hadoop requires the installation of the Java JDK. Although Java 1.5 is the minimum version supported, Apache Hadoop recommends installing the latest version of the Java JDK.

4.3.1 Prerequisites

Download the latest version of the Java JDK from Oracle’s website. The specific commands needed to install and configure the required software may differ between operating systems. All of the command examples provided are for the CentOS operating system.

Install the Java JDK:

```
$sudo yum install sun-java6-jdk
```

To verify the Java JDK installation and version:

```
$java –version
```

Secure Shell (SSH) must be installed on all of the nodes in the cluster and configured with the same port number (default port number is 22). SSH provides management and communication with the nodes in the cluster. To verify the SSH installation:

```
$ssh localhost
```

Apache Hadoop does not currently support IP version 6 (IPv6); therefore, IPv6 must be disabled on the Linux server to ensure only IPv4 is used for IP addressing (this is done by editing the /etc/sysctl.conf file). Edit the /etc/sysctl.conf file:

```
$sudo gedit /etc/sysctl.conf
```

Add the following lines to the end of the /etc/sysctl.conf file:

```
#disable ipv6
net.ipv6.conf.all.disable_ipv6=1
net.ipv6.conf.default.disable_ipv6=1
net.ipv6.conf.lo.disable_ipv6=1
```

Load the /etc/sysctl.conf file settings:

```
$sudo sysctl –f /etc/sysctl.conf
```

To verify IPV6 is disabled:

```
$cat /proc/sys/net/ipv6/conf/all/disable_ipv6
```

If IPv6 is disabled, the above command outputs a 1.

Apache Hadoop requires a dedicated user (hduser) and group (hadoop) to install Apache Hadoop on the system. To create the dedicated user and group for Apache Hadoop:

```
$sudo addgroup hadoop
$sudo adduser --ingroup hadoop hduser
```

4.3.2 Hadoop Installation

Install the current version of Apache Hadoop from Apache's website. Apache recommends installing and configuring Apache Hadoop as a single-node cluster, and once Apache Hadoop is running successfully, configure it as a multi-node cluster.

Download, install and configure (files and environment variables) Apache Hadoop using the installation instructions available on Apache's website. Configure the HDFS file system with the name node. Start the single-node cluster:

```
$sudo /<hadoop folder path>/start-all.sh
```

The installation package provides example programs for testing the Apache Hadoop installation and configuration.

Once the configured single-node cluster is running successfully, configure the multi-node cluster using the installation instructions available on Apache's website. Configure one node in the cluster as the master node, and configure the remaining nodes in the cluster as slave nodes. The master node communicates with the slave nodes to configure and transfer data between the nodes.

Perform the startup of the multi-node cluster in two steps:

- Start the name node daemon on the master node and the data node daemon on the slave nodes
- Start the MapReduce daemons, which starts the jobtracker on the master node and the task tracker on the slave nodes

Stopping the multi-node cluster stops the MapReduce daemons first, and then the name node and data node daemons are stopped [Noll11]. Figure 4 presents an overview of the master and slave nodes in a multi-node cluster.

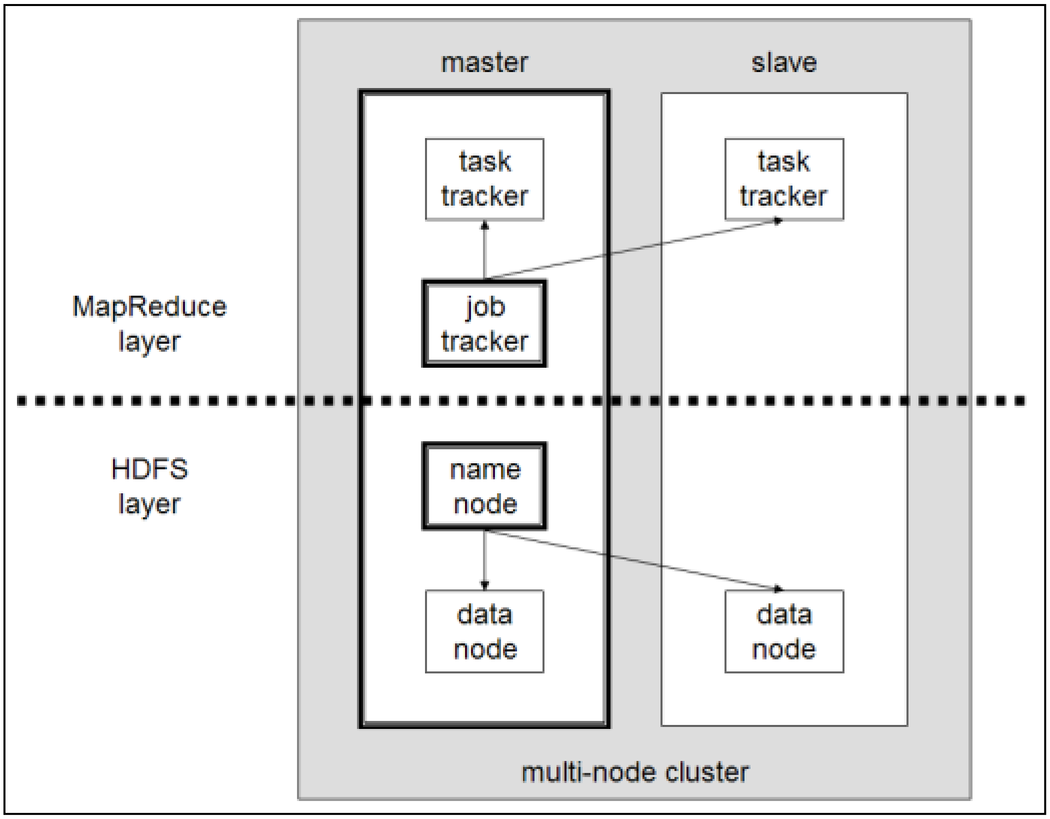

Figure 3: Overview of multi-node cluster [Noll11].

4.4 Sector

Sector supports the Linux and Windows operating system on clients; however, Sector supports only the Linux operating system on the server. Sector requires the installation of the GNU Compiler Collection (gcc) and OpenSSL library. Although gcc version 3.4 is the minimum version supported, Sector recommends installing the latest versions of gcc and OpenSSL.

4.4.1 Sector Installation

Download the current version of Sector from Sector’s project site on the SourceForge website. Once the configured single-node cluster is running successfully, configure the multi-node cluster. Install and configure (files and environment variables) Sector using the installation instructions available on Sector’s project site on the SourceForge website. Start the security server, master and slave nodes.

Start the master and slave nodes of the multi-node cluster:

```
$/<sector-sphere folder path>/start_all
```

Alternatively, start individual master and slave nodes in their respective sub-directory:

```
$/<sector-sphere folder path>/slave-directory.
```

After starting all the nodes, obtain details about the cluster:

```
$/<sector-sphere folder path>/tools/sysinfo
```

The installation package provides tools for verifying and testing the Sector installation and configuration.

Master and slave nodes communicate through the security server using SSH, configured as password-free, for all the nodes. Configure the security server to update all the IP addresses of the master and slave nodes. After all the initial setup is completed, run the startup scripts of security server, master and slave nodes and ensure Sector is up and running. Once Sector is running successfully it can run on the Amazon EC2 or GCE clouds simply by updating the IP address for all the nodes for the respective cloud instances [Gu10B].

Chapter 5

HARDWARE AND SOFTWARE SPECIFICATIONS

5.1 Software Specifications

Install Centos 6.3 version of Linux on the workstations. Install version 1.6 of the Java JDK. Configure SSH on all the nodes on Amazon EC2 and GCE. Install Python on the workstations to run the MalStone scripts. Use versions 2.8 and 0.20.2 for Sector and Hadoop, respectively, to carry out the tests. Use the StarCluster toolkit to create cluster on Amazon EC2. For cluster creation on GCE, install Maven and git from Google's website. Sector installation requires GNU Compiler Collection (gcc) and Open SSL Library, which are available for download from Google's website.

5.2 Hardware Specifications

Use a High-CPU medium instance with moderate I/O performance on the Amazon EC2 cloud service, which includes a memory capacity of 1.7GB and a storage capacity of 350GB. Use a High-CPU-2-d instance on the GCE cloud service, which includes a memory capacity of 3.60GB, a storage capacity of 10GB, and an additional 870GB of scratch disk capacity. Use two Dell workstations with 32 bit Linux OS connected by Gigabit Ethernet cables to perform the research.

Chapter 6

RESULTS AND ANALYSIS

The study evaluates and compares the performance of the Amazon EC2 and GCE cloud services using the TeraSort, MalStone and CreditStone benchmarks, and the p-value obtained through statistical analysis of the collected data using the T-TEST function available in Microsoft Excel 2010. Statistical analysis, resulting in a p-value of less than 0.05, is significant.

The response time (in seconds), for the Amazon EC2 and GCE cloud services are presented in graphs to assist with analyzing trends. The graphs compare the Amazon EC2 and GCE cloud services using each benchmark (TeraSort, MalStone and CreditStone), each distributed file system (Apache Hadoop and Sector), and each dataset size (1GB, 10GB, 100GB and 1TB). For each graph, the y-axis represents the response time values achieved during the tests, and the x-axis represents the number of nodes tested.

6.1 TeraSort Benchmark

6.1.1 Amazon EC2 and GCE Performance on Apache Hadoop

Table 3 and Figures 4 and 5 present the TeraSort benchmark performance response times for Amazon EC2 and GCE using the Apache Hadoop.

Data size	1GB		10GB		100GB		1TB	
#nodes	EC2	GCE	EC2	GCE	EC2	GCE	EC2	GCE
1	40	28	128	106	930	890	5960	5950
2	37	26	122	101	928	875	5951	5946
3	35	25	117	97	912	855	5940	5939
4	32	24	110	92	894	832	5929	5924
5	28	22	100	86	882	812	5914	5910
6	24	19	94	82	870	801	5903	5892
7	21	16	92	75	852	791	5889	5882
8	17	12	86	69	831	779	5876	5860
P-value	0.04		0.02		0.00		0.64	

Table 3: TeraSort – Amazon EC2 vs. GCE on Apache Hadoop.

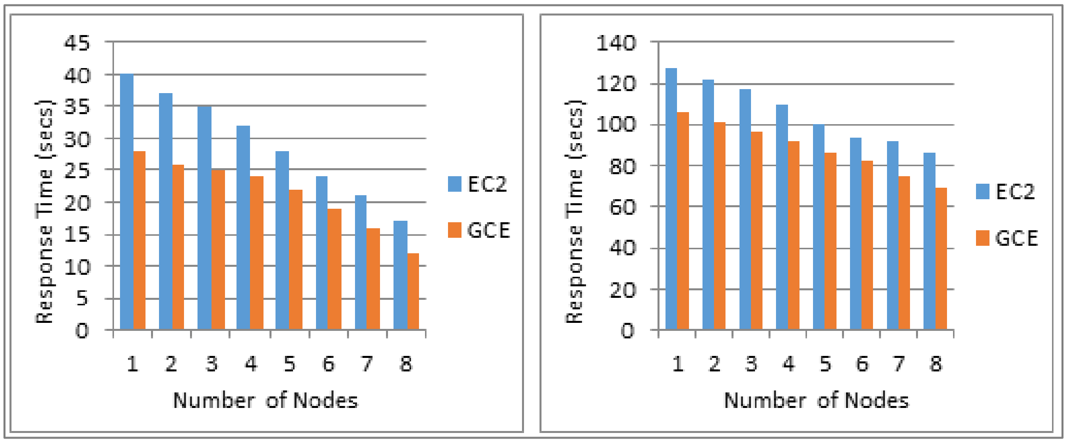

Figure 4: TeraSort – Amazon EC2 vs. GCE on Apache Hadoop (1GB and 10GB).

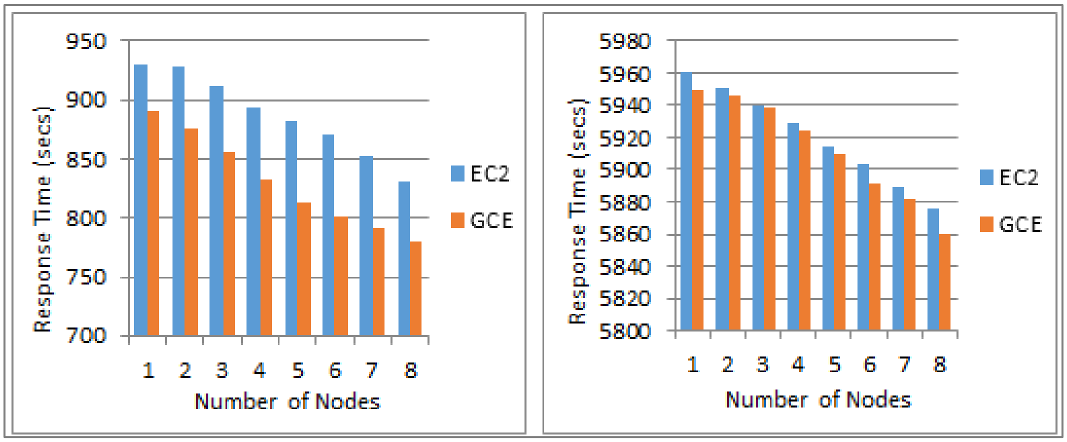

Figure 5: TeraSort – Amazon EC2 vs. GCE on Apache Hadoop (100GB and 1TB).

Statistical analysis with T-Test for the 1GB, 10GB and 100GB TeraSort benchmark test results indicate that the GCE cloud service performed significantly better than the Amazon EC2 cloud service, in terms of response time, using Apache Hadoop; however, the P-value provided by T-Test for the 1TB test results was greater than 0.05 indicating that the difference in the result set for GCE and EC2 was statistically insignificant.

6.1.2 Amazon EC2 and GCE Performance on Sector

Table 4 and Figures 6 and 7 present the TeraSort benchmark performance response times for Amazon EC2 and GCE using the Sector.

Data size	1GB		10GB		100GB		1TB	
#nodes	EC2	GCE	EC2	GCE	EC2	GCE	EC2	GCE
1	30	18	66	41	480	440	2990	2960
2	26	16	64	38	472	432	2981	2947
3	22	13	55	34	457	426	2970	2934
4	19	11	51	30	444	419	2959	2920
5	15	9	47	24	437	411	2941	2901
6	12	7	42	21	429	401	2930	2890
7	10	5	38	18	418	389	2922	2841
8	8	4	29	16	401	351	2892	2776
P-value	0.04		0.00		0.02		0.05	

Table 4: TeraSort – Amazon EC2 vs. GCE on Sector.

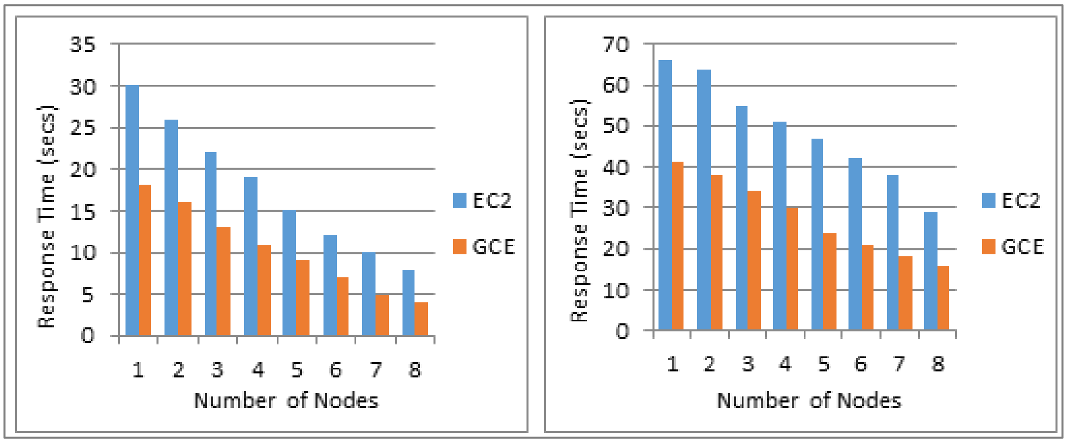

Figure 6: TeraSort – Amazon EC2 vs. GCE on Sector (1GB and 10GB).

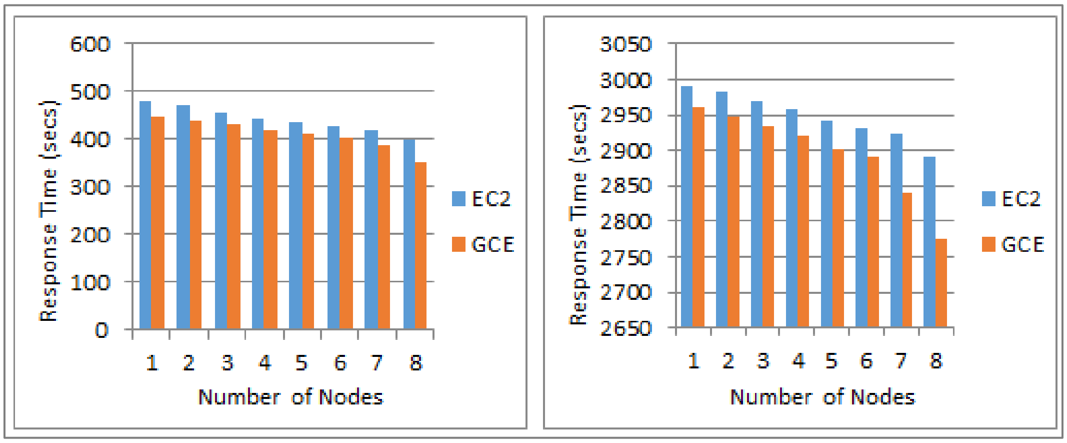

Figure 7: TeraSort – Amazon EC2 vs. GCE on Sector (100GB and 1TB).

Statistical analysis with T-Test for the 1GB, 10GB and 100GB TeraSort benchmark test results indicate that the GCE cloud service performed significantly better than the Amazon EC2 cloud service, in terms of response time, using Sector; however, the P-value provided by T-Test for the 1TB test results was greater than 0.05 indicating that the difference in the result set for GCE and EC2 was statistically insignificant.

6.1.3 Apache Hadoop and Sector Performance on Amazon EC2

Table 5 and Figures 8 and 9 present the TeraSort benchmark performance response times for Apache Hadoop and Sector on Amazon EC2.

Data size	1GB		10GB		100GB		1TB	
# nodes	Hadoop	Sector	Hadoop	Sector	Hadoop	Sector	Hadoop	Sector
1	40	30	128	66	930	480	5960	2990
2	37	26	122	64	928	472	5951	2981
3	35	22	117	55	912	457	5940	2970
4	32	19	110	51	894	444	5929	2959
5	28	15	100	47	882	437	5914	2941
6	24	12	94	42	870	429	5903	2930
7	21	10	92	38	852	418	5889	2922
8	17	8	86	29	831	401	5876	2892
P-value	0.01		0.00		0.00		0.00	

Table 5: TeraSort – Apache Hadoop vs. Sector on Amazon EC2.

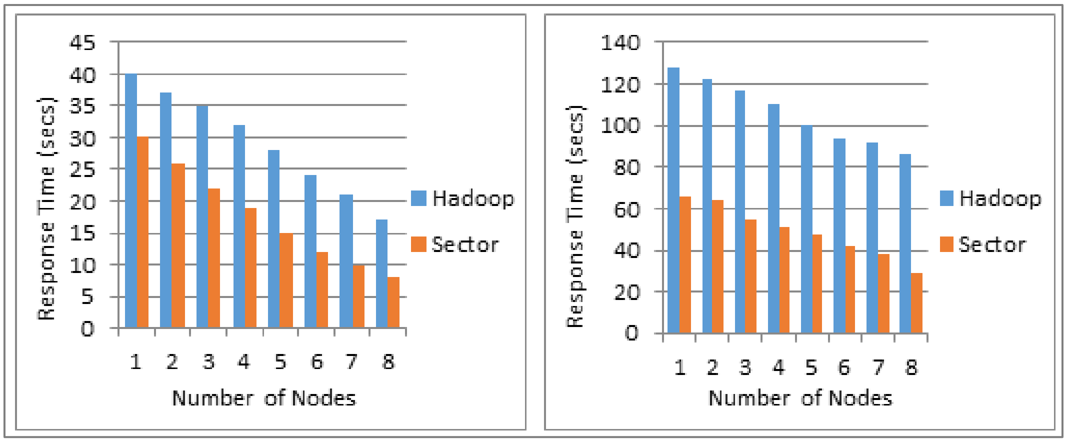

Figure 8: TeraSort – Apache Hadoop vs. Sector on Amazon EC2 (1GB and 10GB).

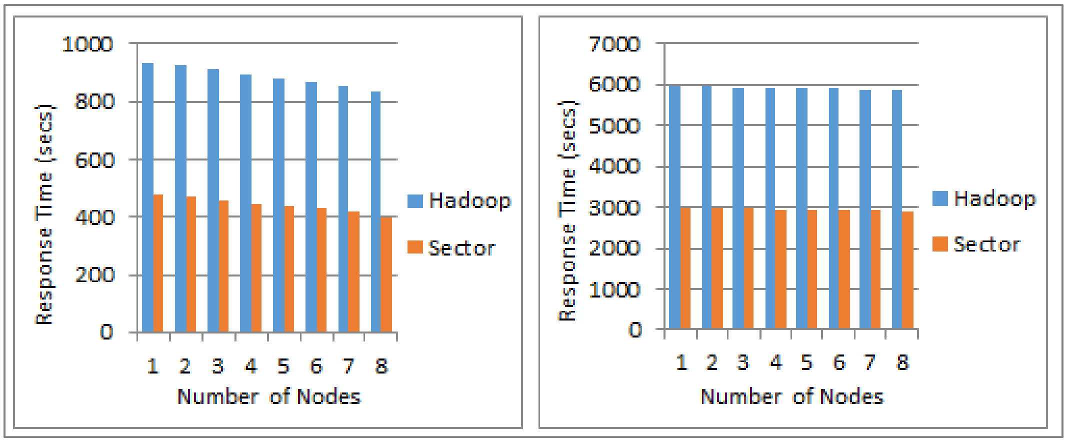

Figure 9: TeraSort – Apache Hadoop vs. Sector on Amazon EC2 (100GB and 1TB).

6.1.4 Apache Hadoop and Sector Performance on GCE

Table 6 and Figures 10 and 11 present the TeraSort benchmark performance response times for Apache Hadoop and Sector on GCE.

Data size	1GB		10GB		100GB		1TB	
# nodes	Hadoop	Sector	Hadoop	Sector	Hadoop	Sector	Hadoop	Sector
1	28	18	106	41	890	440	4920	2960
2	26	16	101	38	875	432	4918	2947
3	25	13	97	34	855	426	4915	2934
4	24	11	92	30	832	419	4907	2920
5	22	9	86	24	812	411	4901	2901
6	19	7	82	21	801	401	4890	2890
7	16	5	75	18	791	389	4882	2841
8	12	4	69	16	779	351	4870	2776
P-value	**0.00**		**0.00**		**0.00**		**0.00**	

Table 6: TeraSort – Apache Hadoop vs. Sector on GCE.

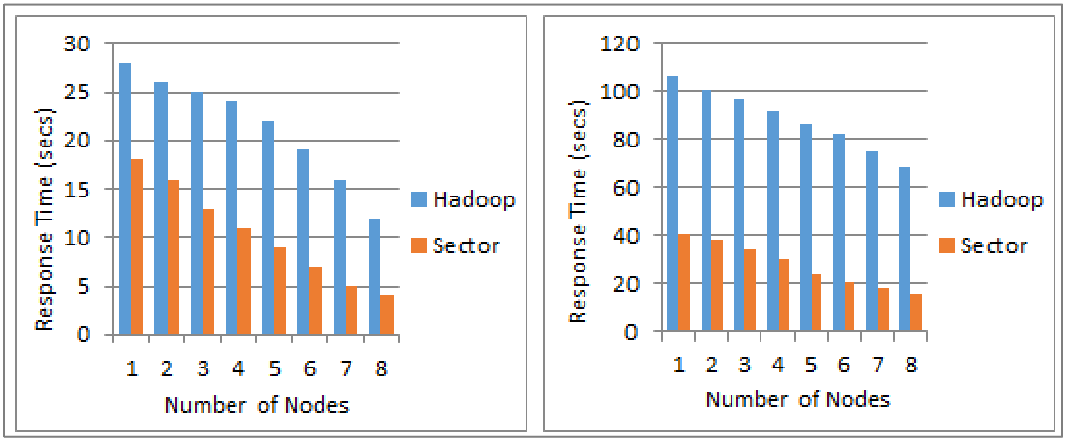

Figure 10: TeraSort – Apache Hadoop vs. Sector on GCE (1GB and 10GB).

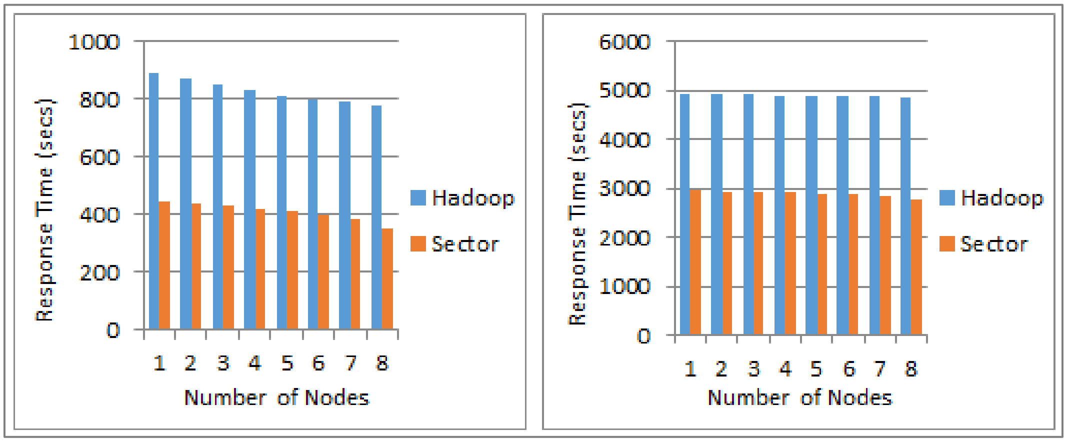

Figure 11: TeraSort – Apache Hadoop vs. Sector on GCE (100GB and 1TB).

6.2 CreditStone Benchmark

6.2.1 Amazon EC2 and GCE Performance on Apache Hadoop

Table 7 and Figures 12 and 13 present the CreditStone benchmark performance response times for Amazon EC2 and GCE using Apache Hadoop.

Data Size	1GB		10GB		100GB		1TB	
# nodes	EC2	GCE	EC2	GCE	EC2	GCE	EC2	GCE
1	40	29	101	90	450	420	3100	3090
2	37	26	96	86	443	410	3085	3081
3	34	24	92	80	439	398	3072	3062
4	30	21	88	75	434	392	3061	3054
5	26	17	83	71	428	386	3058	3046
6	22	13	79	64	419	377	3040	3032
7	19	10	72	57	405	368	3010	3001
8	13	8	68	50	392	349	2980	2970
P-value	0.05		0.05		**0.00**		0.66	

Table 7: CreditStone – Amazon EC2 vs. GCE on Apache Hadoop.

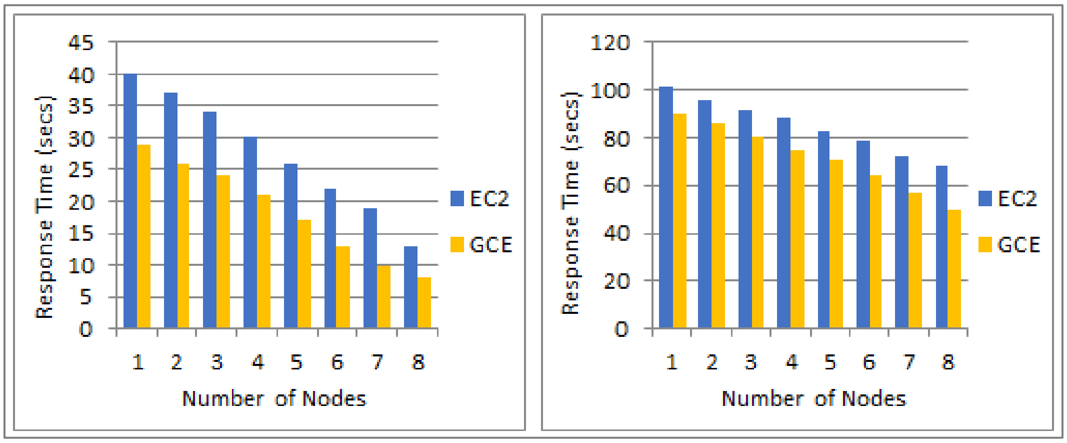

Figure 12: CreditStone – Amazon EC2 vs. GCE on Apache Hadoop (1GB and 10GB).

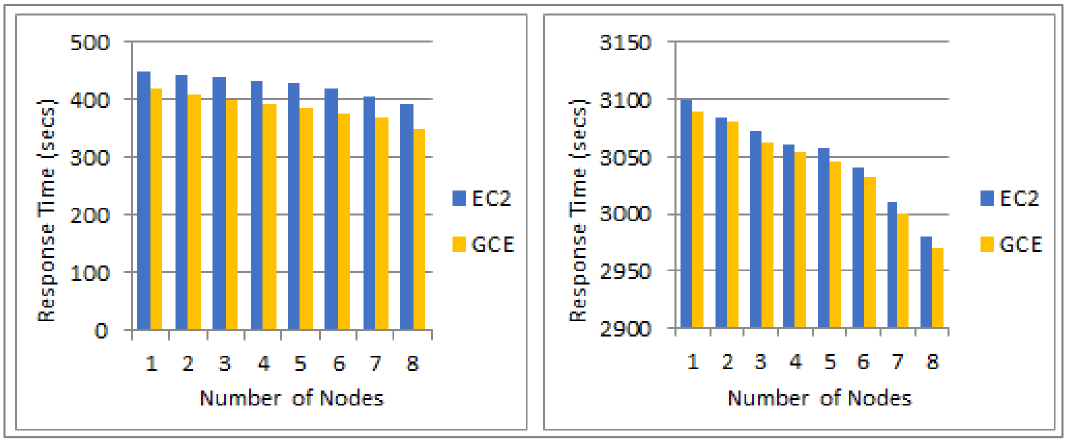

Figure 13: CreditStone – Amazon EC2 vs. GCE on Apache Hadoop (100GB and 1TB).

Statistical analysis with T-Test for the 100GB CreditStone benchmark test results indicate that the GCE cloud service performed significantly better than the Amazon EC2 cloud service, in terms of response time, using Apache Hadoop; however, the P-value provided by T-Test for the 1GB, 10GB and 1TB test results were greater than 0.05 indicating that the difference in the result set for GCE and EC2 was statistically insignificant.

6.2.2 Amazon EC2 and GCE Performance on Sector

Table 8 and Figures 14 and 15 present the CreditStone benchmark performance response times for Amazon EC2 and GCE using Sector.

Data Size	1GB		10GB		100GB		1TB	
# nodes	EC2	GCE	EC2	GCE	EC2	GCE	EC2	GCE
1	30	16	53	44	250	210	1560	1546
2	24	14	49	39	238	207	1555	1542
3	21	12	46	36	222	196	1548	1539
4	17	11	43	32	212	189	1540	1531
5	13	9	41	28	200	180	1530	1525
6	11	6	37	24	192	173	1512	1518
7	10	5	29	19	183	161	1500	1508
8	7	2	24	10	171	142	1479	1489
P-value	**0.04**		0.05		0.05		0.79	

Table 8: CreditStone – Amazon EC2 vs. GCE on Sector.

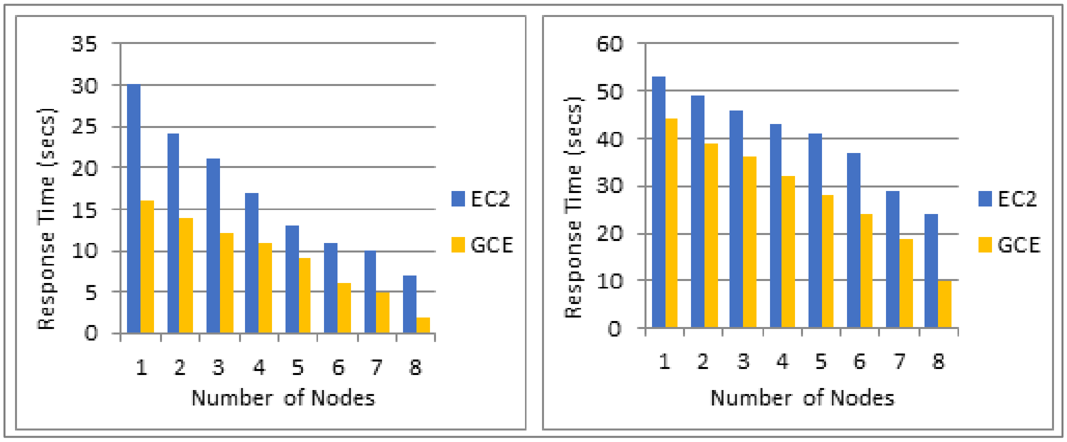

Figure 14: CreditStone – Amazon EC2 vs. GCE on Sector (1GB and 10GB).

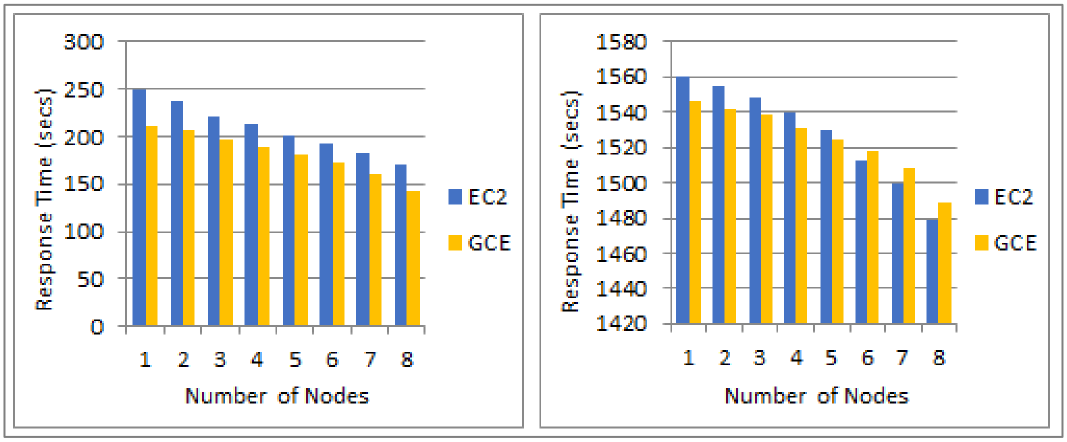

Figure 15: CreditStone – Amazon EC2 vs. GCE on Sector (100GB and 1TB).

Statistical analysis with T-Test for the 1GB CreditStone benchmark test results indicate that the GCE cloud service performed significantly better than the Amazon EC2 cloud service, in terms of response time, using Sector; however, the P-value provided by T-Test for the 10GB, 100GB and 1TB test results was greater than 0.05 indicating that the difference in the result set for GCE and EC2 was statistically insignificant.

6.2.3 Apache Hadoop and Sector Performance on Amazon EC2

Table 9 and Figures 16 and 17 present the CreditStone benchmark performance response times for Apache Hadoop and Sector on Amazon EC2.

Data Size	1GB		10GB		100GB		1TB	
# nodes	Hadoop	Sector	Hadoop	Sector	Hadoop	Sector	Hadoop	Sector
1	40	30	101	53	450	250	3100	1560
2	37	24	96	49	443	238	3085	1555
3	34	21	92	46	439	222	3072	1548
4	30	17	88	43	434	212	3061	1540
5	26	13	83	41	428	200	3058	1530
6	22	11	79	37	419	192	3040	1512
7	19	10	72	29	405	183	3010	1500
8	13	7	68	24	392	171	2980	1479
P-value	**0.02**		**0.00**		**0.00**		**0.00**	

Table 9: CreditStone – Apache Hadoop vs. Sector on Amazon EC2.

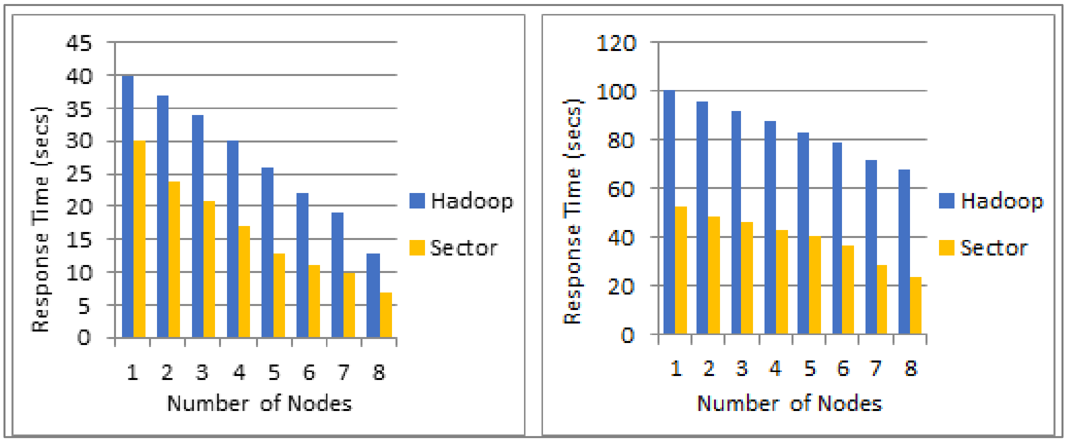

Figure 16: CreditStone – Apache Hadoop vs. Sector on Amazon EC2 (1GB and 10GB).

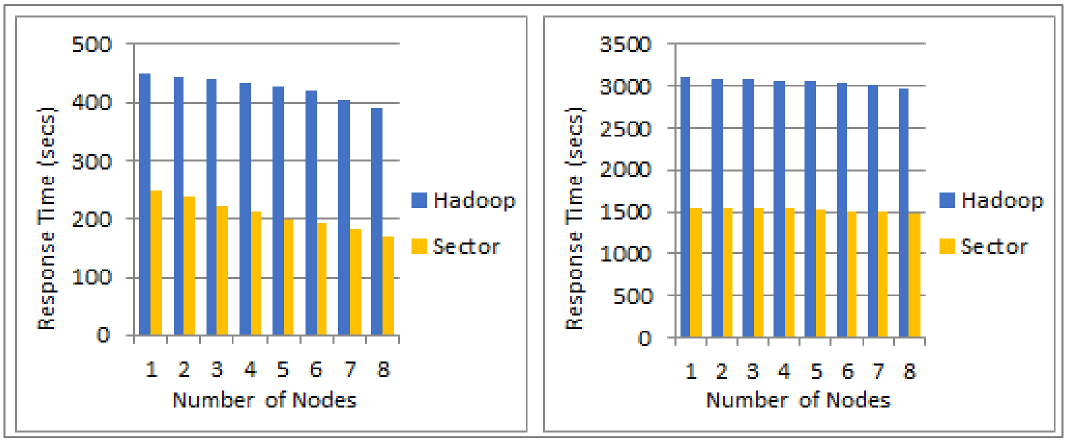

Figure 17: CreditStone – Apache Hadoop vs. Sector on Amazon EC2 (100GB and 1TB).

6.2.4 Apache Hadoop and Sector Performance on GCE

Table 10 and Figures 18 and 19 present the CreditStone benchmark performance response times for Apache Hadoop and Sector on GCE.

Data Size	1GB		10GB		100GB		1TB	
# nodes	Hadoop	Sector	Hadoop	Sector	Hadoop	Sector	Hadoop	Sector
1	29	16	90	44	420	210	3090	1546
2	26	14	86	39	410	207	3081	1542
3	24	12	80	36	398	196	3062	1539
4	21	11	75	32	392	189	3054	1531
5	17	9	71	28	386	180	3046	1525
6	13	6	64	24	377	173	3032	1518
7	10	5	57	19	368	161	3001	1508
8	8	2	50	10	349	142	2970	1489
P-value	**0.01**		**0.00**		**0.00**		**0.00**	

Table 10: CreditStone – Apache Hadoop vs. Sector on GCE.

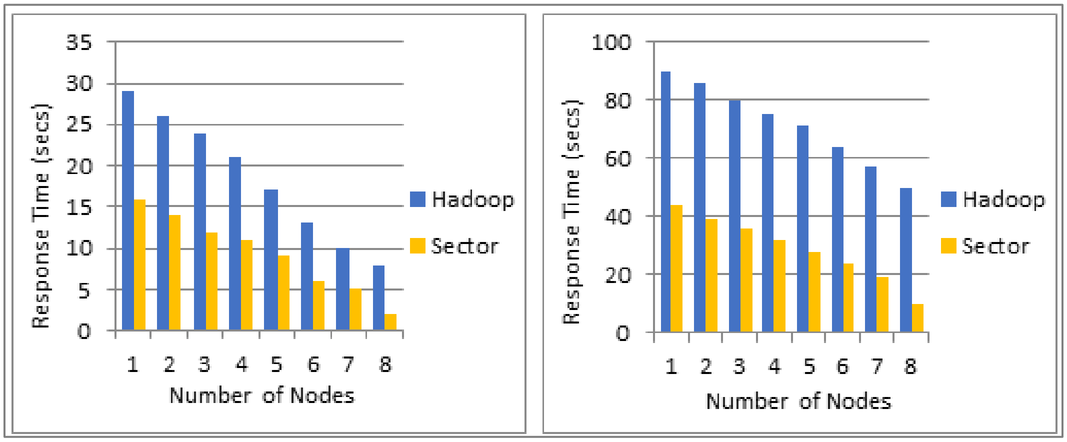

Figure 18: CreditStone – Apache Hadoop vs. Sector on GCE (1GB and 10GB).

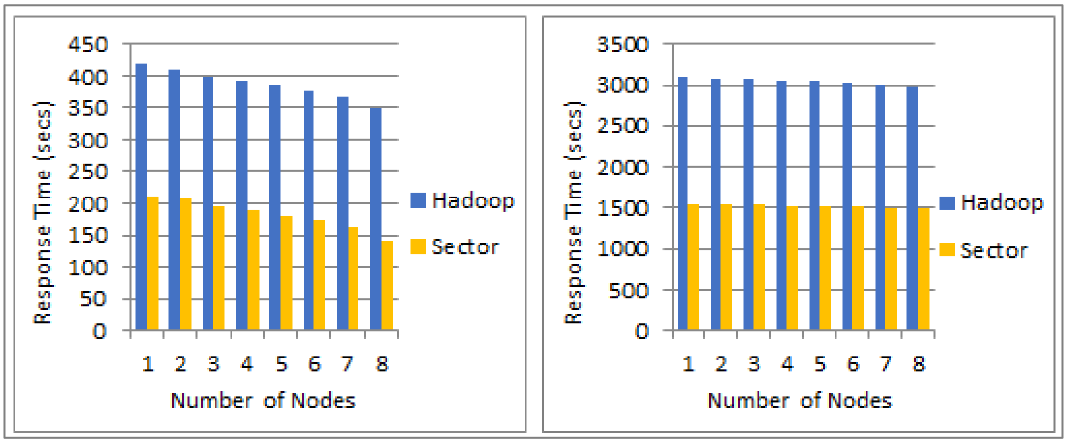

Figure 19: CreditStone – Apache Hadoop vs. Sector on GCE (100GB and 1TB).

6.3 MalStone Benchmark

6.3.1 Amazon EC2 and GCE Performance on Apache Hadoop

Table 11 and Figures 20 and 21 present the MalStoneA benchmark performance response times for Amazon EC2 and GCE using Apache Hadoop.

Data Size	1GB		10GB		100GB		1TB	
# nodes	EC2	GCE	EC2	GCE	EC2	GCE	EC2	GCE
1	40	31	120	109	527	517	4870	4861
2	35	28	116	105	519	508	4862	4852
3	31	24	112	103	513	501	4852	4844
4	29	20	107	98	503	495	4841	4838
5	26	18	104	94	496	487	4835	4831
6	22	12	101	88	488	482	4826	4822
7	19	9	97	82	476	471	4821	4816
8	17	8	89	70	463	460	4816	4809
P-value	0.05		0.05		**0.44**		0.51	

Table 11: MalStoneA – Amazon EC2 and GCE on Apache Hadoop.

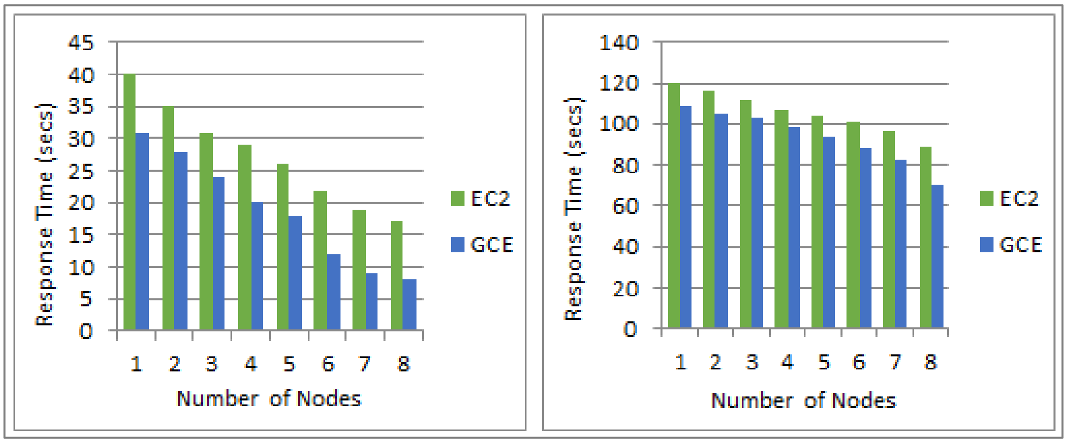

Figure 20: MalStoneA – Amazon EC2 vs. GCE on Apache Hadoop (1GB and 10GB).

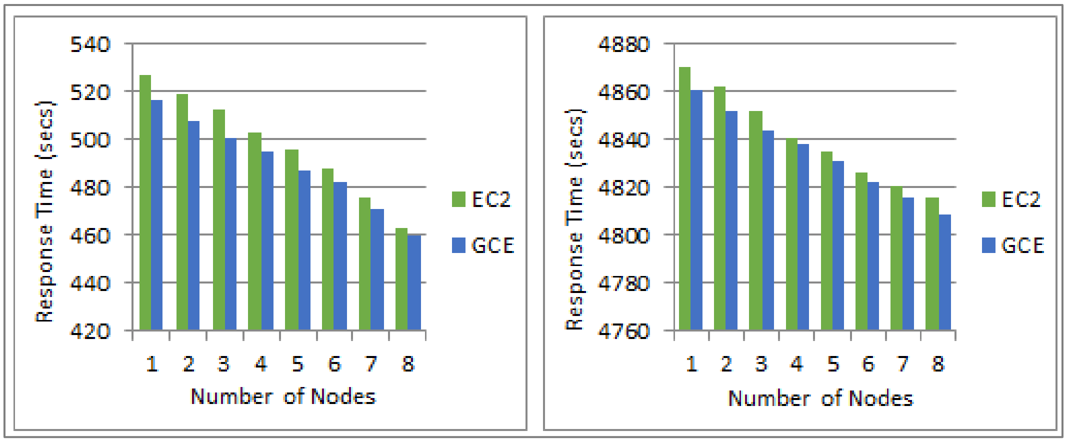

Figure 21: MalStoneA – Amazon EC2 vs. GCE on Apache Hadoop (100GB and 1TB).

Statistical analysis with T-Test for the 1GB, 10GB, 100GB and 1TB MalStoneA benchmark test results was greater than 0.05 indicating that the difference in the result set for GCE and EC2 was statistically insignificant.

Table 12 and Figures 22 and 23 present the MalStoneB benchmark performance response times for Amazon EC2 and GCE using Apache Hadoop.

Data Size	1GB		10GB		100GB		1TB	
# nodes	EC2	GCE	EC2	GCE	EC2	GCE	EC2	GCE
1	80	72	340	329	1030	1028	8715	8710
2	78	68	335	321	1028	1024	8712	8704
3	75	59	329	314	1025	1020	8708	8697
4	72	51	323	303	1020	1015	8705	8691
5	69	49	319	292	1015	1008	8700	8686
6	66	44	311	288	1005	997	8696	8678
7	62	42	302	273	990	989	8688	8659
8	58	40	292	258	980	970	8678	8651
P-value	**0.00**		0.05		0.59		0.09	

Table 12: MalStoneB – Amazon EC2 vs. GCE on Apache Hadoop.

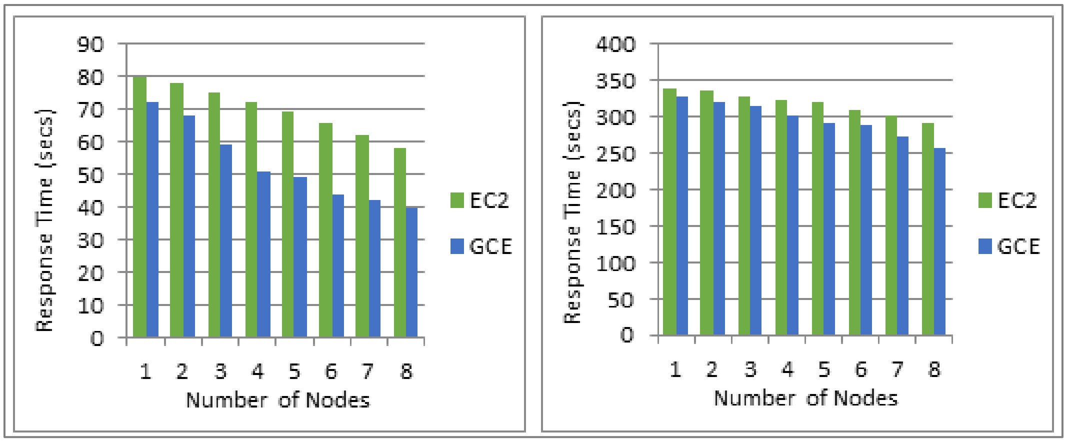

Figure 22: MalStoneB – Amazon EC2 vs. GCE on Apache Hadoop (1GB and 10GB).

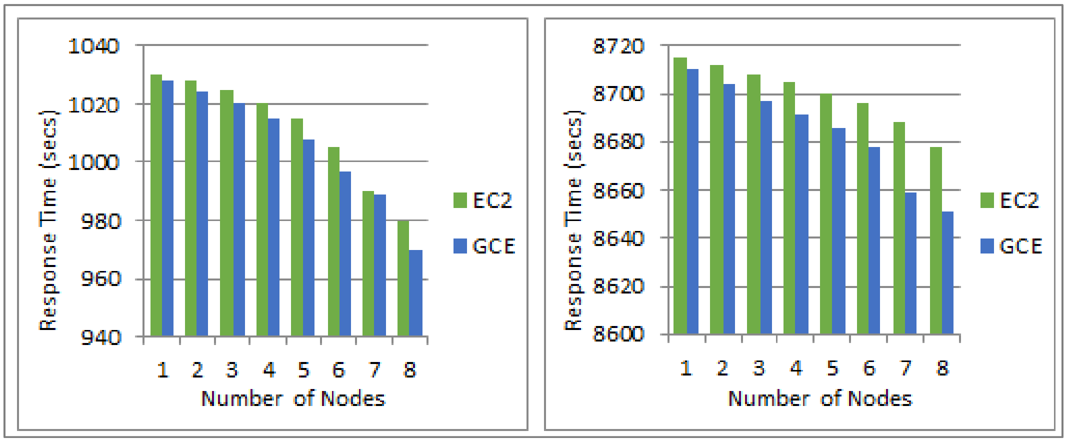

Figure 23: MalStoneB – Amazon EC2 vs. GCE on Apache Hadoop (100GB and 1TB).

Statistical analysis with T-Test for the 1GB MalStoneB benchmark test results indicate that the GCE cloud service performed significantly better than the Amazon EC2 cloud service, in terms of response time, using Apache Hadoop; however, the P-value provided by T-Test for the 10GB, 100GB and 1TB test results was greater than 0.05 indicating that the difference in the result set for GCE and EC2 was statistically insignificant.

6.3.2 Amazon EC2 and GCE Performance on Sector

Table 13 and Figures 24 and 25 present the MalStoneA benchmark performance response times for Amazon EC2 and GCE using Sector.

Data Size	1GB		10GB		100GB		1TB	
# nodes	EC2	GCE	EC2	GCE	EC2	GCE	EC2	GCE
1	25	17	60	52	265	245	2440	2425
2	22	15	58	49	260	240	2431	2415
3	20	13	56	45	251	220	2425	2402
4	18	10	52	42	240	198	2422	2392
5	15	9	50	38	231	184	2414	2386
6	11	8	48	35	220	178	2409	2376
7	9	5	44	32	210	178	2394	2368
8	8	3	36	27	199	178	2385	2357
P-value	0.05		**0.02**		**0.02**		**0.03**	

Table 13: MalStoneA – Amazon EC2 vs. GCE on Sector.

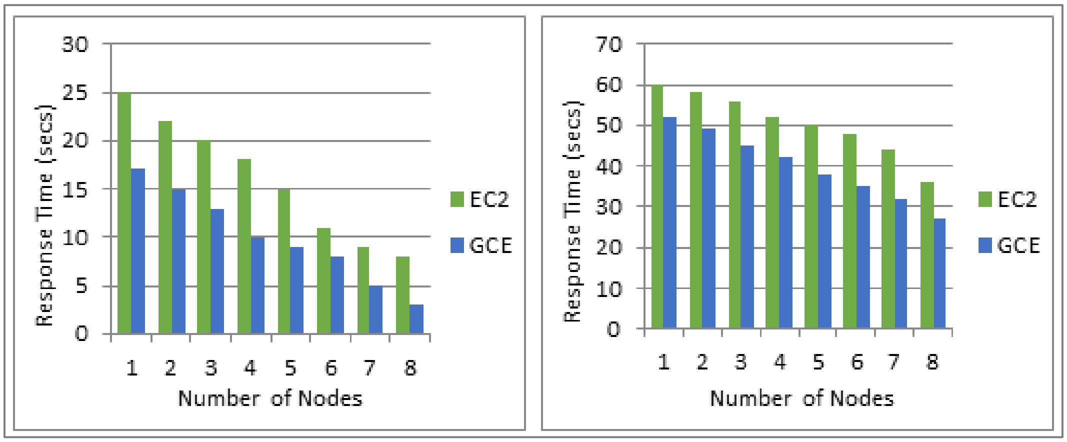

Figure 24: MalStoneA – Amazon EC2 vs. GCE on Sector (1GB and 10GB).

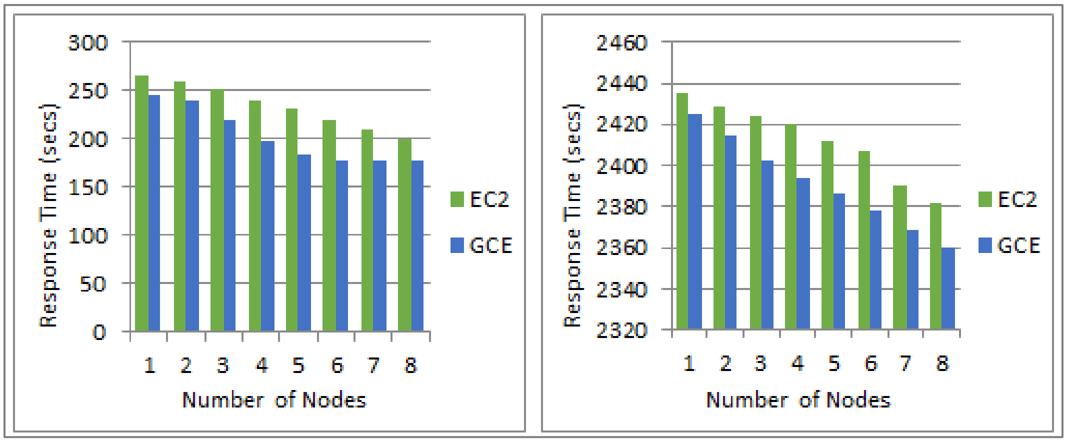

Figure 25: MalStoneA – Amazon EC2 vs. GCE on Sector (100GB and 1TB).

Statistical analysis with T-Test for the 10GB, 100GB and 1TB MalStoneA benchmark test results indicate that the GCE cloud service performed significantly better than the Amazon EC2 cloud service, in terms of response time, using Sector; however, the P-value provided by T-Test for the 1GB test results was greater than 0.05 indicating that the difference in the result set for GCE and EC2 was statistically insignificant.

Table 14 and Figures 26 and 27 present the MalStoneB benchmark performance response times for Amazon EC2 and GCE using Sector.

Data Size	1GB		10GB		100GB		1TB	
# nodes	EC2	GCE	EC2	GCE	EC2	GCE	EC2	GCE
1	42	30	172	160	520	500	4350	4335
2	40	27	167	150	509	489	4342	4320
3	36	25	163	138	503	470	4339	4306
4	31	22	154	126	497	455	4333	4292
5	28	21	150	118	488	442	4325	4285
6	25	19	142	108	480	436	4318	4276
7	20	13	136	96	475	427	4307	4262
8	18	11	128	84	464	418	4300	4250
P-value	**0.03**		**0.02**		**0.01**		**0.01**	

Table 14: MalStoneB – Amazon EC2 vs. GCE on Sector.

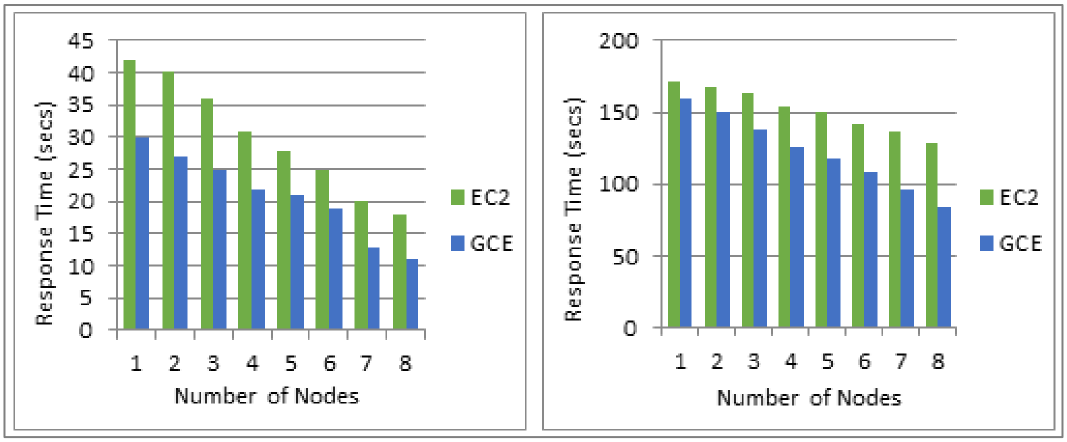

Figure 26: MalStoneB – Amazon EC2 vs. GCE on Sector (1GB and 10GB).

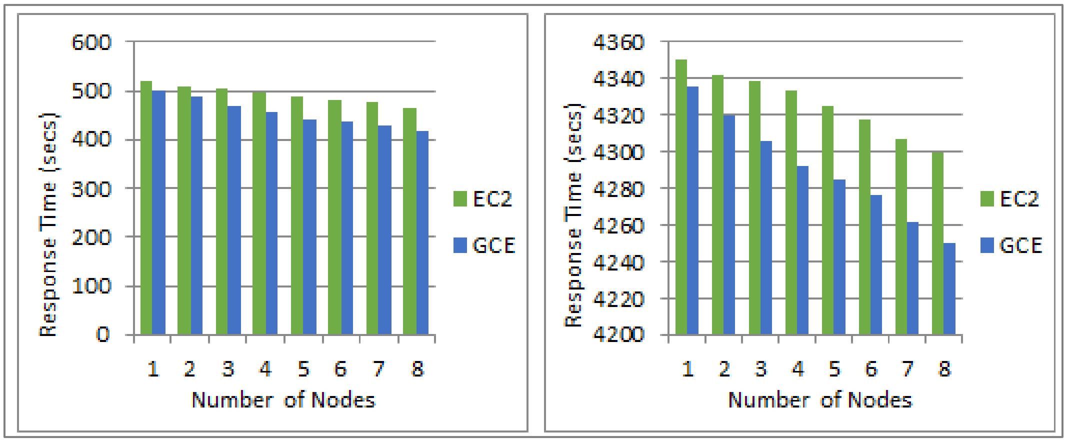

Figure 27: MalStoneB – Amazon EC2 vs. GCE on Sector (100GB and 1TB).

The MalStoneB benchmark test results indicate that the GCE cloud service performed significantly better than the Amazon EC2 cloud service, in terms of response time, using Sector.

6.3.3 Apache Hadoop vs. Sector Performance on Amazon EC2

Table 15 and Figures 28 and 29 present the MalStoneA benchmark performance response times for Apache Hadoop and Sector on Amazon EC2.

Data Size	1GB		10GB		100GB		1TB	
# nodes	Hadoop	Sector	Hadoop	Sector	Hadoop	Sector	Hadoop	Sector
1	40	25	120	60	527	265	4870	2440
2	35	22	116	58	519	260	4862	2431
3	31	20	112	56	513	251	4852	2425
4	29	18	107	52	503	240	4841	2422
5	26	15	104	50	496	231	4835	2414
6	22	11	101	48	488	220	4826	2409
7	19	9	97	44	476	210	4821	2394
8	17	8	89	36	463	199	4816	2385
P-value	**0.00**		**0.00**		**0.00**		**0.00**	

Table 15: MalStoneA – Apache Hadoop vs. Sector on Amazon EC2.

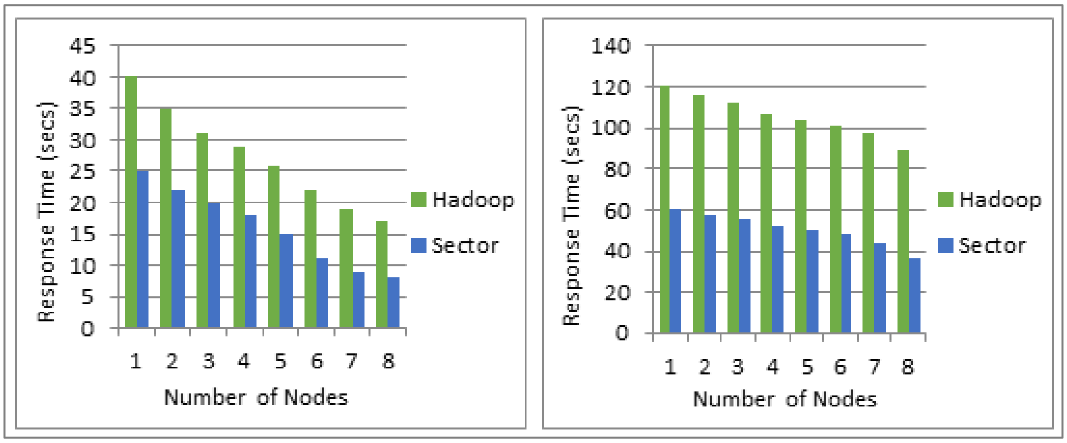

Figure 28: MalStoneA – Apache Hadoop vs. Sector on Amazon EC2 (1GB and 10GB).

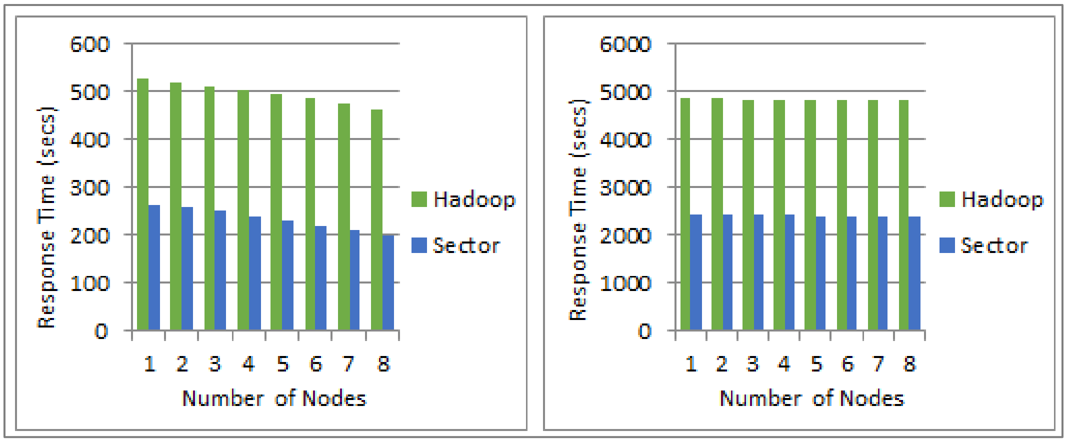

Figure 29: MalStoneA – Apache Hadoop vs. Sector on Amazon EC2 (100GB and 1TB).

Table 16 and Figures 30 and 31 present the MalStoneB benchmark performance response times for Apache Hadoop and Sector on Amazon EC2 and GCE.

Data Size	1GB		10GB		100GB		1TB	
# nodes	Hadoop	Sector	Hadoop	Sector	Hadoop	Sector	Hadoop	Sector
1	80	42	340	172	1030	520	8715	4350
2	78	40	335	167	1028	509	8712	4342
3	75	36	329	163	1025	503	8708	4339
4	72	31	323	154	1020	497	8705	4333
5	69	28	319	150	1015	488	8700	4325
6	66	25	311	142	1005	480	8696	4318
7	62	20	302	136	990	475	8688	4307
8	58	18	292	128	980	464	8678	4300
P-value	**0.00**		**0.00**		**0.00**		**0.00**	

Table 16: MalStoneB – Apache Hadoop vs. Sector on Amazon EC2.

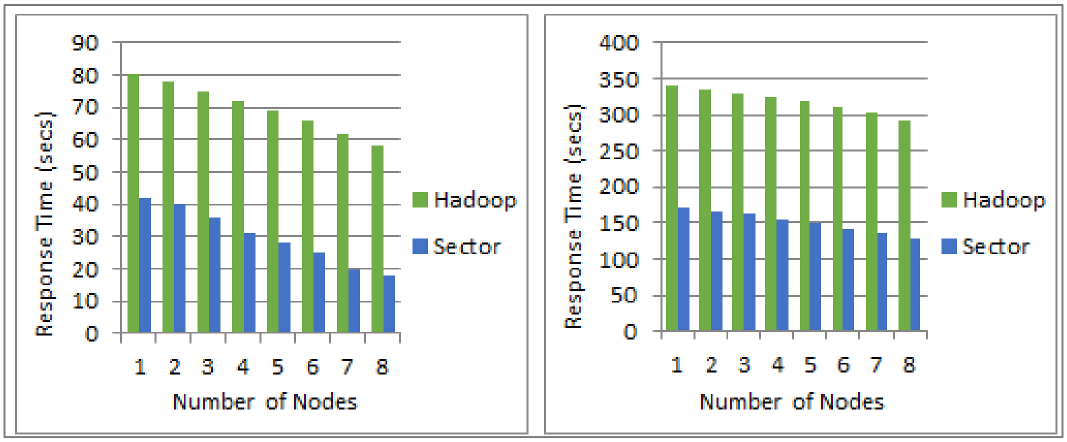

Figure 30: MalStoneB – Apache Hadoop vs. Sector on Amazon EC2 (1GB and 10GB).

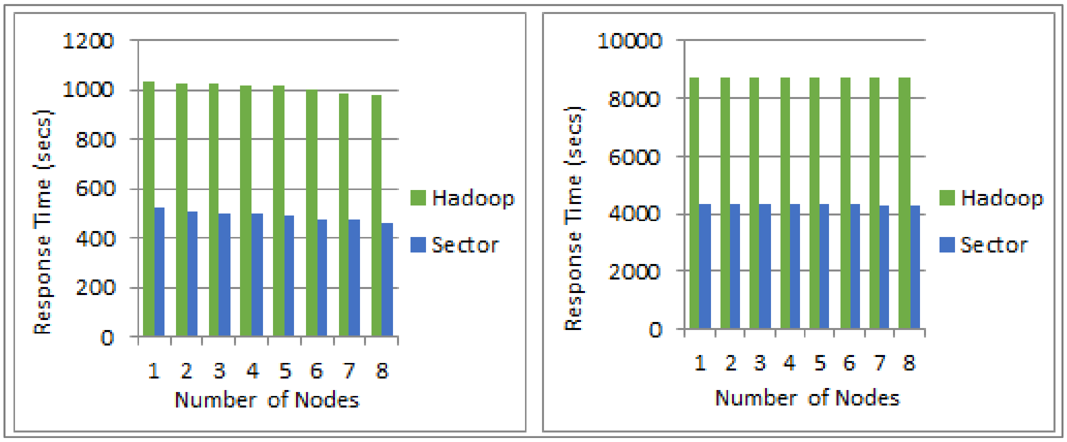

Figure 31: MalStoneB – Apache Hadoop vs. Sector on Amazon EC2 (100GB and 1TB).

6.3.4 Apache Hadoop vs. Sector Performance on GCE

Table 17 and Figures 32 and 33 present the MalStoneA benchmark performance response times for Apache Hadoop and Sector on GCE.

Data Size	1GB		10GB		100GB		1TB	
# nodes	Hadoop	Sector	Hadoop	Sector	Hadoop	Sector	Hadoop	Sector
1	31	17	109	52	517	245	4861	2425
2	28	15	105	49	508	240	4852	2415
3	24	13	103	45	501	220	4844	2402
4	20	10	98	42	495	198	4838	2392
5	18	9	94	38	487	184	4831	2386
6	12	8	88	35	482	178	4822	2376
7	9	5	82	32	471	178	4816	2368
8	8	3	70	27	460	178	4809	2357
P-value	**0.02**		**0.00**		**0.00**		**0.00**	

Table 17: MalStoneA – Apache Hadoop vs. Sector on GCE.

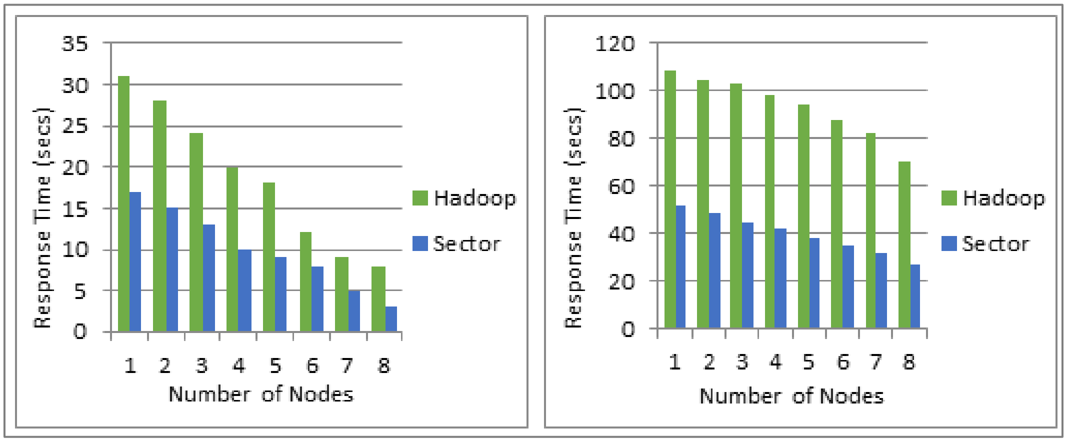

Figure 32: MalStoneA – Apache Hadoop vs. Sector on GCE (1GB and 10GB).

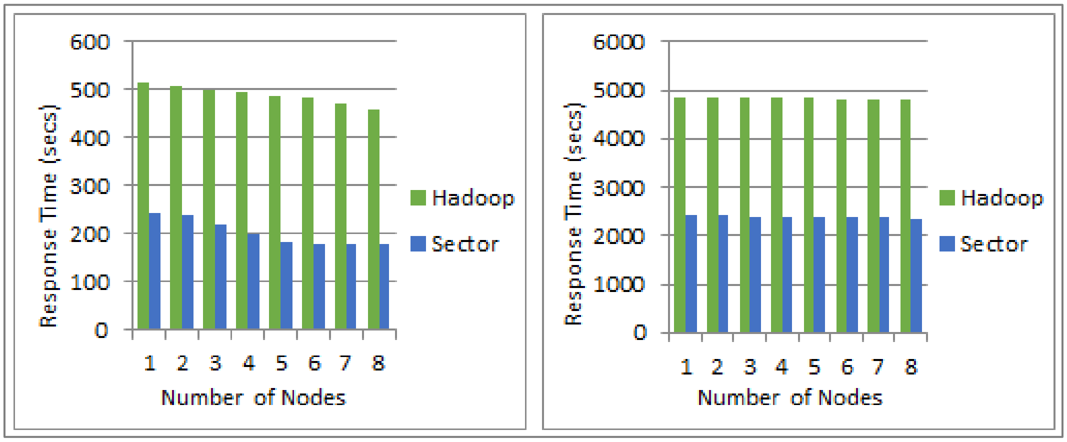

Figure 33: MalStoneA – Apache Hadoop vs. Sector on GCE (100GB and 1TB).

Table 18 and Figures 34 and 35 present the MalStoneB benchmark performance response times for Apache Hadoop and Sector on GCE.

Data Size	1GB		10GB		100GB		1TB	
# nodes	Hadoop	Sector	Hadoop	Sector	Hadoop	Sector	Hadoop	Sector
1	78	30	329	160	1028	500	8710	4335
2	75	27	321	150	1024	489	8704	4320
3	71	25	314	138	1020	470	8697	4306
4	68	22	303	126	1015	455	8691	4292
5	66	21	292	118	1008	442	8686	4285
6	62	19	288	108	997	436	8678	4276
7	55	13	273	96	989	427	8659	4262
8	49	11	258	84	970	418	8651	4250
P-value	**0.00**		**0.00**		**0.00**		**0.00**	

Table 18: MalStoneB – Apache Hadoop vs. Sector on GCE.

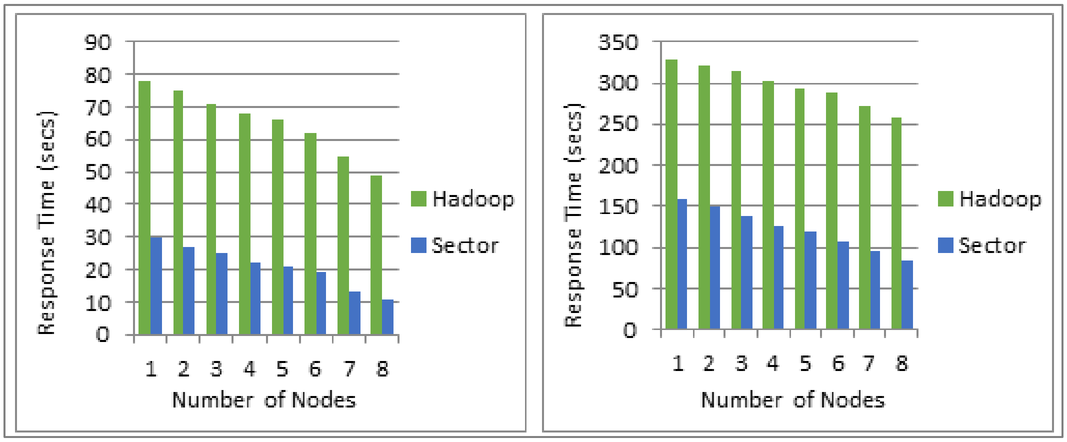

Figure 34: MalStoneB – Apache Hadoop vs. Sector on GCE (1GB and 10GB).

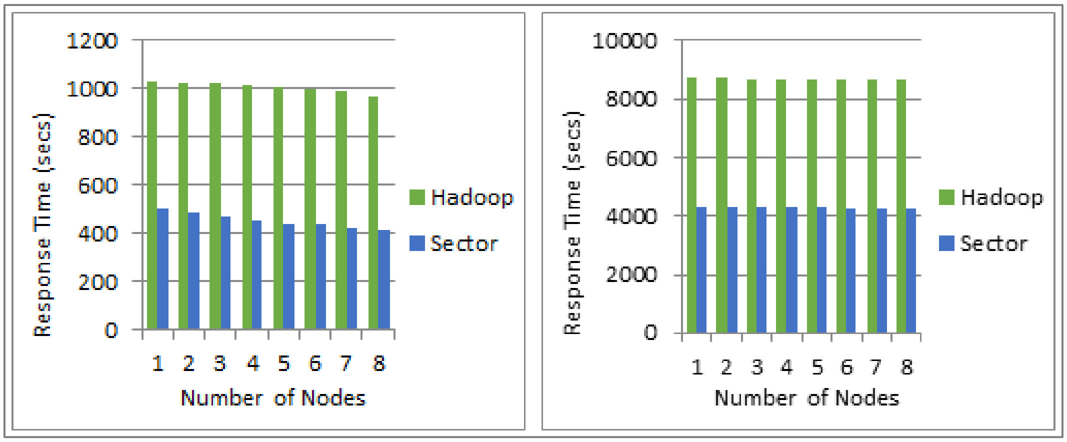

Figure 35: MalStoneB – Apache Hadoop vs. Sector on GCE (100GB and 1TB).

Chapter 7

CONCLUSIONS

The following is a discussion of the results presented in chapter 6, organized by the benchmarks studied. The Amazon EC2 and GCE cloud services were tested using the Apache Hadoop and Sector/Sphere distributed file systems while the number of nodes (1 to 8) and the size of the dataset (1GB, 10GB, 100GB and 1TB) were varied. Although this study focused on the performance of the Amazon EC2 and GCE cloud services using Apache Hadoop and Sector/Sphere, control tests were performed to verify that the Apache Hadoop and Sector/Sphere distributed file systems, tested on the Amazon EC2 (tables 5, 9, 15-16 and figures 8-9, 16-17, 28-31) and GCE (tables 6, 10, 17-18 and figures 10-11, 18-19, 32-35), performed as well as the results presented on Apache Hadoop (tables 3, 7, 11-12 and figures 4-5, 12-13, 20-23) and on Sector (tables 4, 8, 13-14 and figures 6-7, 14-15, 24-27) on the Amazon EC2 and GCE cloud services. The control tests show that on the Amazon EC2 cloud service, Sector/Sphere produced response time results that were, on average, 2.04 times better than Apache Hadoop, and on the GCE cloud service, Sector/Sphere produced response time results that were, on average, 2.34 times better than Apache Hadoop, which concurs with the results of earlier research.

7.1 Benchmark Results

7.1.1 TeraSort

The goal of the TeraSort benchmark is to sort 1TB of data, or any other amount of data, as fast as possible. It is a benchmark that combines testing the HDFS and MapReduce layers of a Hadoop cluster and allows us to compare results across clouds. Apache Hadoop produced significantly better response times on the GCE cloud service compared to the Amazon EC2 cloud service using the 1GB, 10GB and 100GB datasets; however, there was no significant difference between the two cloud services using the 1TB dataset. Sector/Sphere produced significantly better response times on the GCE cloud service compared to the Amazon EC2 cloud service using the 1GB, 10GB and 100GB datasets; however, there was no significant difference between the two cloud services using the 1TB datasets. The 1TB data showed that when it comes to big data, Amazon EC2 and GCE performed about the same and in this case the end user should consider other factors like the services offered and the pricing model of the public clouds.

Unlike GCE cloud, the high response times on Amazon EC2 cloud might be due to the external storage on the cloud and the cloud's moderate network performance. Amazon EC2 also experience longer instance startup times compared to GCE.

7.1.2 CreditStone

The Creditstone benchmark, modeled to test the clouds that provide on demand computing capacity, is used mainly by banking applications. Apache Hadoop only produced significantly better response times on the GCE cloud service compared to the Amazon EC2 cloud service using the 100GB dataset while there was no significant difference between the two cloud services using the 1GB, 10GB and 1TB datasets. Sector only produced significantly better response times on the GCE cloud service compared to the Amazon EC2 cloud service using the 1GB dataset while there was no significant difference between the two cloud services using the 10GB, 100GB and 1TB datasets.

CreditStone benchmark, executed on Amazon EC2 and GCE clouds, showed significant performance for all the data sizes with Hadoop having high response times compared to Sector because Hadoop uses block level processing while Sector uses file level processing, which can greatly reduce processing time. GCE accounted for better response times due to the features of efficient disk performance for the read/write operations and storage disks mounted on the instances directly. The instance boot up time was less on GCE compared to Amazon EC2. Persistent disks on GCE showed consistent high performance.

7.1.3 MalStone

MalStone, a stylized benchmark is widely used for cyber security applications. The benchmark is executed on Amazon EC2 and GCE clouds for varying data sets from 1GB to 1TB in two sets i.e., MalStone A tested for the data obtained for a day and MalStone B tested for the data obtained for a week. Using the MalStone-A datasets, Apache Hadoop produced no significant difference between the two cloud services, and Sector produced significantly better response times on the GCE cloud service compared to the Amazon EC2 cloud service using the 10GB, 100GB and 1TB datasets while there was no significant difference between the two cloud services using the 1GB dataset.

Using the MalStone-B datasets, Apache Hadoop only produced significantly better response times on the GCE cloud service compared to the Amazon EC2 cloud service using the 1GB dataset while there was no significant difference between the two cloud services using the 10GB, 100GB and 1TB datasets, and Sector produced significantly better response times on the GCE cloud service compared to the Amazon EC2 cloud service. This is likely due to the delay incurred by the use of block level processing by Hadoop, which divides the data into blocks before processing the data. This behavior added to Amazon EC2's storage system accounted for further poor performance on Amazon EC2 cloud with higher response times compared to GCE cloud service. MalStone A and MalStone B tested on Amazon EC2 and GCE clouds showed significant performance with Hadoop and Sector.

7.2 Hadoop vs. Sector

The response times for the data generation on a single node is more than the time on group of nodes due to the data shared between the nodes in case of cluster and the time taken to process the data on each node is relatively less. Sector implemented file level processing which accepts the data generated by the benchmark in the form of files and there is no further division of data into smaller chunks. Sector also makes use of Filesystem in Userspace (FUSE) interface, which helps to mount the file system directly on to the operating system. The feature of data locality on Sector results in reduced data transfer between the nodes. Sector integrates data storage and processing into a single system that each storage node uses to process the data.

Hadoop, which is widely used for data intensive applications, showed poor response times on Amazon EC2 cloud and from a pure performance perspective, Hadoop on EC2 is not as good an option as the storage on the cloud is remote and the delays in the write operation. Though EC2 performed equally well with GCE cloud on read operations, poor write operations overcome the performance. Sector addresses a limitation in HDFS, which prevents mounting a file system directly on the existing OS.

GCE, a perfect alternative for Amazon EC2 in IaaS layer of the cloud showed better performance compared to Amazon EC2. The features provided by the cloud like the memory, storage, compute units and the virtual cores were almost the same on both the clouds. But GCE addressed the limitations faced by Amazon EC2 cloud and resulted in more compute power for lesser investments on the cloud. GCE showed blazing performance for the various data loads compared to Amazon EC2. Table 19 shows the summary of the response times and the significant performance of the cloud services Amazon EC2 and GCE. GCE represents the cloud's significant performance compared to Amazon EC2 for the respective data size and the statistically insignificant performance between the clouds is represented by same. Sector showed significant performance compared to Apache Hadoop for all the data sizes and for the three benchmarks.

Benchmark	Hadoop				Sector				Hadoop Vs. Sector
	1GB	10GB	100GB	1TB	1GB	10GB	100GB	1TB	
TeraSort	GCE	GCE	GCE	Same	GCE	GCE	GCE	Same	Sector
CreditStone	Same	Same	GCE	Same	GCE	Same	Same	Same	Sector
MalStone-A	Same	Same	GCE	Same	Same	GCE	GCE	GCE	Sector
MalStone-B	GCE	Same	Same	Same	GCE	GCE	GCE	GCE	Sector

Table 19: Summary table for performance results

7.3 Future Research

This study is limited to benchmarking the Amazon EC2 and GCE cloud services to evaluate the performance on data intensive applications while varying workloads and the number of nodes in the cluster.

Extensions to this study on cloud performance include evaluating operating systems other than Linux such as Windows and Apple OS X, evaluating additional cloud services such as Microsoft's Windows Azure, evaluating a wider range of nodes within the cluster, evaluating additional benchmarks, and measuring additional performance metrics. Since research involving the GCE cloud service is lacking, this study can serve as a reference to future studies involving data intensive computing on the GCE cloud service involving data.

REFERENCES

Print Publications:

[Bennett10]
Bennett, Collin, R.L.Grossman, D. Locke, J. Seidman and S. Vejcik, “MalStone: Towards a Benchmark for Analytics on Large Data Clouds”, Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, New York, 2010, pp. 145-152.

[Grossman08]
Grossman, R.L., C. Bennett, and J. Seidman, “CreditStone: A Benchmark for Clouds That Provide On-Demand Capacity”, University of Illinois at Chicago, Chicago, Illinois, 2008, pp. 638-643.

[Grossman09A]
Grossman, R.L., Y. Gu, M. Sabala, and W. Zhang, “Compute and Storage Clouds Using Wide Area High Performance Networks”, Journal Future Generation Computer Systems 25, 2 (February, 2009), ACM, New York, New York, pp. 179-183.

[Grossman09B]
Grossman, R.L. and Y. Gu, “On the Varieties of Clouds for Data Intensive Computing”, Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 32, 2009, pp. 44-50.

[Gu09]
Gu, Y. and R.L. Grossman, “Sector and Sphere: Towards Simplified Storage and Processing of Large Scale Distributed Data”, Philosophical Transactions of the Royal Society A, 367, 2009, pp. 2429-2445.

[Gu10A]
Gu, Y. and R.L. Grossman, "Towards Efficient and Simplified Distributed Data Intensive Computing", IEEE Transactions on Parallel and Distributed Systems (TPDS), 22, 2011, pp. 974-984.

[Jogalekar00]
Jogalekar P. and M. Woodside, "Evaluating the Scalability of Distributed Systems," IEEE Transactions on Parallel and Distributed Systems, 11, 6 (2000), pp. 589-603.

Electronic Sources:

[AWS13]
"What is an EC2 Compute Unit and Why Did You Introduce It?", http://aws.amazon.com/ec2/faqs/#What_is_an_EC2_Compute_Unit_and_why_did_you_introduce_it, last accessed July 13, 2013.

[Gu10B]
Gu, Y., "Sector Installation and Usage Tutorial", http://sector.sourceforge.net/pub/sector-tutorial-0710-v2.pdf, last revision July 2010, last accessed July 13, 2013.

[Hadoop13]
"Apache Hadoop: MapReduce", http://developer.yahoo.com/hadoop/tutorial/module4.html, last accessed July 13, 2013.

[O'Malley08]
O'Malley, O., "Terabyte Sort on Apache Hadoop", http://www.sortbenchmark.org/YahooHadoop.pdf, last revision May 2008, last accessed July 13, 2013.

[NIST11]
Brown, E., "Final Version of NIST Cloud Computing Definition Published", National Institute of Standards and Technology, http://www.nist.gov/itl/csd/cloud-102511.cfm, last revision October 25, 2011, last accessed July 13, 2013.

[Noll11]
Noll, M.G., "Benchmarking and Stress Testing a Hadoop Cluster With Terasort, TestDFSIO & Co.", http://www.michael-noll.com/blog/2011/04/09/benchmarking-and-stress-testing-an-hadoop-cluster-with-terasort-testdfsio-nnbench-mrbench/, last revision April 9, 2011, last accessed July 13, 2013.

[Pervasive10]
"Pervasive", http://cs.pervasive.com/blogs/datarush/archive/2010/03/05/cluster-on-a-chip.aspx, last revision March 5, 2010, last accessed July 13, 2013.

[StarCluster13]
"What is StarCluster?", http://web.mit.edu/star/cluster/docs/latest/overview.html, last accessed July 13, 2013.

Appendix A

MALSTONE BENCHMARK

The description of each field is as below:

Event ID- This is a sequential identifier for each record. The script distinguishes datasets across the nodes by adding a hash to the hostname, which is a non-numeric value, while generating the datasets. Length of Event ID is 31 bytes.

Timestamp- This 26-byte field is a random value distributed uniformly and is used to denote the time and date of the event. The default value is 365 days.

SiteID- This identifies the site associated with the event. Field width is 19 bytes.

Flag- This is 1-byte value field used to indicate a compromised site. Compromised sites are marked with a value of 1, and uncompromised sites with a flag value of 0.

Entity ID- This is a 19-byte identifier of an entity associated with the event.

Appendix B

Cluster setup on Amazon EC2

StarCluster 0.93.3V is available for download from StarCluster web site [StarCluster 12]. Set up the configuration file and fill in the AWS credentials and key pair info. Update the default cluster template with the name of EC2 key pair. Start the cluster mycluster using the command

```
$ starcluster start mycluster
```

Start working on the cluster by logging into the master node as root by running the command

```
$ starcluster sshmaster mycluster
```

Copy the files to the cluster from local system to the cluster using the command

```
$ starcluster put /path/to/local/file/or/dir /remote/path
```

To copy files from cluster to local machine, use the command

```
$ starcluster get /path/to/local/file/or/dir /local/path
```

Terminate the cluster by running the command

```
$ starcluster terminate mycluster
```

Appendix C

Cluster setup on GCE

Launching the cluster involves the following steps:

1. One-Time setup

Grant the required permissions and run the one-time setup using the commands

```
$ chmod +x one_time_seup.sh
$ ./one-time-setup.sh
```

2. Set up firewalls

This step is to allow tools to run on local workstation to contact and communicate with the coordinator node. The default port set in cfg.py is 8888.

```
$ gcutil –project=<project> addfirewall snitch –
description=”Let coordinator and snitches chatter.” –
allowed=”tcp: 8888”
```

3. Hardcode the config files

Edit the config file to change the zone, machine type, image, and disk properties in tools/launch_coordinator.py. Set the bucket size and project ID in tools/common.py. Edit the environment variables for Hadoop and Sector in the env.sh files. Set the bucket to:

```
$ gsutil mb gs://bucket_name
```

4. Regular operation

This step launches the slave nodes in the cluster. Google storage allows the sharing of common configurations and scripts. To launch the cluster, run:

```
$ ./tools/launch_coordinator.py
$ ./tools/begin_hadoop.py num_slaves for Hadoop
$ ./tools/begin_sector.py num_slaves for Sector slave
nodes
```

VITA

Bhagavathi Kaza earned her undergraduate degree in Electrical and Electronics Engineering from Jawaharlal Nehru Technological University (JNTU), Hyderabad, India. She is currently pursuing a master's degree of Computers and Information Sciences at the University of North Florida. She currently works as a Software Engineer for Johnson & Johnson, a leading company for Contact Lens Manufacturing in Jacksonville, Florida.