

2017

## Empirical Evaluation of Cloud IAAS Platforms using System-level Benchmarks

Niharika Deval

*University of North Florida*, n00892706@ospreys.unf.eduFollow this and additional works at: <https://digitalcommons.unf.edu/etd> Part of the [Digital Communications and Networking Commons](#)

---

### Suggested Citation

Deval, Niharika, "Empirical Evaluation of Cloud IAAS Platforms using System-level Benchmarks" (2017).  
*UNF Graduate Theses and Dissertations*. 765.  
<https://digitalcommons.unf.edu/etd/765>

This Master's Thesis is brought to you for free and open access by the Student Scholarship at UNF Digital Commons. It has been accepted for inclusion in UNF Graduate Theses and Dissertations by an authorized administrator of UNF Digital Commons. For more information, please contact [Digital Projects](#).  
© 2017 All Rights Reserved

EMPIRICAL EVALUATION OF CLOUD IAAS PLATFORMS USING  
SYSTEM-LEVEL BENCHMARKS

by

Niharika Deval

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

Fall, 2017

Copyright © 2017 by Niharika Deval

All rights reserved. Reproduction in whole or in part in any form requires the prior written permission of Niharika Deval or designated representative.

The thesis "Empirical Evaluation of Cloud IaaS Platforms using System-Level Benchmarks" submitted by Niharika Deval 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. Asai Asaithambi

Accepted for the School of Computing:

---

Dr. Sherif Elfayoumy  
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. John Kantner  
Dean of the Graduate School

## ACKNOWLEDGEMENT

I wish to thank my parents, Gopal Rao Deval and Jyothi Deval, and my brother, Navin Kumar Deval, for their unwavering support and concern during the many hours I dedicated to accomplishing this milestone in my life and career. I highly appreciate the understanding and patience of my family while I achieved one of the greatest moments of my life. I also wish to extend my gratitude to my friend Rajendra Prasad Arukala for sharing his knowledge and encouraging me throughout this period.

I thank my thesis advisor Dr. Sanjay P. Ahuja for his expert advice, feedback, and suggestions. I also thank Dr. Roger Eggen and Dr. Asai Asaithambi who agreed to be on my thesis committee and provided great feedback in this research process. Working with these three distinguished UNF faculty members has been an honor and a privilege.

## CONTENTS

List of Figures .....	viii
List of Tables .....	ix
Abstract .....	xi
Chapter 1: Introduction .....	- 1 -
1.1 Standardization and Transparency .....	- 2 -
1.2 Performance Evaluation and Benchmarking.....	- 3 -
1.3 Cloud Architectures.....	- 5 -
1.3.1 Amazon EC2 .....	- 5 -
1.3.2 Google Compute Engine .....	- 6 -
1.3.3 Microsoft Azure .....	- 7 -
1.3.4 Rackspace Cloud Services .....	- 9 -
1.4 Research Objectives .....	- 10 -
Chapter 2: Literature Review .....	- 12 -
2.1 Studies on Performance and Performance Variability .....	- 12 -
2.2 Studies on Pricing.....	- 14 -
Chapter 3: Research Methodology.....	- 16 -
3.1 Server Performance .....	- 16 -
3.2 File I/O Performance .....	- 17 -
3.3 Network Performance .....	- 18 -
3.4 Performance Variability .....	- 19 -

3.5	Pricing .....	- 20 -
Chapter 4: Metrics and Specifications .....		- 21 -
4.1	Metrics.....	- 21 -
4.2	Hardware and Software Specifications .....	- 23 -
4.2.1	Hardware Specifications .....	- 23 -
4.2.2	Software Specifications .....	- 24 -
Chapter 5: Test Methodology .....		- 25 -
5.1	Performance Benchmarking and Performance Variability .....	- 26 -
5.1.1	Server Performance.....	- 26 -
5.1.2	File I/O Performance.....	- 28 -
5.1.3	Network Performance .....	- 30 -
5.2	Price-per-Performance Analysis .....	- 31 -
5.2.1	Price-per-Server Performance.....	- 31 -
5.2.2	Price-per-File I/O Performance.....	- 32 -
5.2.3	Price-per-Network Performance .....	- 32 -
Chapter 6: Results and Analysis .....		- 33 -
6.1	Server Performance .....	- 34 -
6.1.1	General Purpose Instances .....	- 34 -
6.1.2	Memory-Intensive Instances.....	- 38 -
6.1.3	Compute-Intensive Instances .....	- 42 -
6.2	File I/O Performance .....	- 46 -
6.2.1	General Purpose Instances .....	- 46 -
6.2.2	Memory-Intensive Instances.....	- 51 -

6.2.3	Compute-Intensive Instances .....	- 55 -
6.3	Network Performance .....	- 59 -
6.3.1	General Purpose Instances .....	- 59 -
6.3.2	Memory-Intensive Instances .....	- 62 -
6.3.3	Compute-Intensive Instances .....	- 65 -
Chapter 7: Conclusions and Future Work.....		- 69 -
7.1	Further Research .....	- 70 -
References.....		- 72 -
Appendix A: Unixbench Installation and Execution .....		- 77 -
Appendix B: Dbench Installation and Execution.....		- 80 -
Appendix C: Iperf Installation and Execution .....		- 83 -
Appendix D: Statistical Analysis .....		- 86 -
Vita.....		- 94 -



## FIGURES

Figure 1: Single-Threaded Scores on General Purpose Instance .....	35 -
Figure 2: Multi-Threaded Scores on General Purpose Instance .....	35 -
Figure 3: Single-Threaded Scores on Memory-Intensive Instance.....	39 -
Figure 4: Multi-Threaded Scores on Memory-Intensive Instance.....	39 -
Figure 5: Single-Threaded Scores on Compute-Intensive Instance.....	43 -
Figure 6: Multi-Threaded Scores on Compute-Intensive Instance .....	43 -
Figure 7: Standard File I/O Throughput on General Purpose Instance .....	47 -
Figure 8: File I/O Overhead Throughput on General Purpose Instance .....	47 -
Figure 9: Standard File I/O Throughput on Memory-Intensive Instance .....	51 -
Figure 10: File I/O Overhead Throughput on Memory-Intensive Instance.....	52 -
Figure 11: Standard File I/O Throughput on Compute-Intensive Instance .....	56 -
Figure 12: File I/O Overhead Throughput on Compute-Intensive Instance .....	56 -
Figure 13: Network Throughput on General Purpose Instance .....	60 -
Figure 14: Network Throughput on Memory-Intensive Instance .....	63 -
Figure 15: Network Throughput on Compute-Intensive Instance .....	66 -

## TABLES

Table 1: Description of Amazon EC2 Instance Types.....	6 -
Table 2: Description of Google Compute Engine Machine Types.....	7 -
Table 3: Description of Microsoft Azure Virtual Machines.....	9 -
Table 4: Description of Rackspace Cloud Servers .....	10 -
Table 5: General Purpose Instances across Cloud Providers.....	23 -
Table 6: Memory-Intensive Instances across Cloud Providers.....	24 -
Table 7: Compute-Intensive Instances across Cloud Providers.....	24 -
Table 8: Single- and Multi-Threaded Scores on General Purpose Instance .....	35 -
Table 9: Single-Threaded Scores with CoV on General Purpose Instance .....	37 -
Table 10: Multi-Threaded Scores with CoV on General Purpose Instance.....	37 -
Table 11: Price-per-Server Performance on General Purpose Instance.....	38 -
Table 12: Single- and Multi-Threaded Scores on Memory-Intensive Instance.....	38 -
Table 13: Single-Threaded Scores with CoV on Memory-Intensive Instance .....	41 -
Table 14: Multi-Threaded Scores with CoV on Memory-Intensive Instance.....	41 -
Table 15: Price-per-Server Performance on Memory-Intensive Instance .....	42 -
Table 16: Single- and Multi-Threaded Scores on Compute-Intensive Instance.....	42 -
Table 17: Single-Threaded Scores with CoV on Compute-Intensive Instance .....	45 -
Table 18: Multi-Threaded Scores with CoV on Compute-Intensive Instance.....	45 -
Table 19: Price-per-Server Performance on Compute-Intensive Instance .....	46 -
Table 20: File I/O (STD and OHD) Throughput on General Purpose Instance .....	47 -

Table 21: Standard File I/O Throughput with CoV on General Purpose Instance .....	49 -
Table 22: File I/O Overhead Throughput with CoV on General Purpose Instance.....	50 -
Table 23: Price-per-File I/O Performance on General Purpose Instance .....	50 -
Table 24: File I/O (STD and OHD) Throughput on Memory-Intensive Instance .....	51 -
Table 25: Standard File I/O Throughput with CoV on Memory-Intensive Instance....	53 -
Table 26: File I/O Overhead Throughput with CoV on Memory-Intensive Instance...	54 -
Table 27: Price-per-File I/O Performance on Memory-Intensive Instance .....	54 -
Table 28: File I/O (STD and OHD) Throughput on Compute-Intensive Instance .....	55 -
Table 29: Standard File I/O Throughput with CoV on Compute-Intensive Instance ...	58 -
Table 30: File I/O Overhead Throughput with CoV on Compute-Intensive Instance..	58 -
Table 31: Price-per-File I/O Performance on Compute-Intensive Instance .....	59 -
Table 32: Network Throughput on General Purpose Instance.....	59 -
Table 33: Network Throughput with CoV on General Purpose Instance.....	61 -
Table 34: Price-per-Network Performance on General Purpose Instance .....	62 -
Table 35: Network Throughput on Memory-Intensive Instance .....	62 -
Table 36: Network Throughput with CoV on Memory-Intensive Instance.....	64 -
Table 37: Price-per-Network Performance on Memory-Intensive Instance.....	65 -
Table 38: Network Throughput on Compute-Intensive Instance .....	65 -
Table 39: Network Throughput with CoV on Compute-Intensive Instance.....	67 -
Table 40: Price-per-Network Performance on Compute-Intensive Instance.....	68 -

## ABSTRACT

Cloud Computing is an emerging paradigm in the field of computing where scalable IT enabled capabilities are delivered ‘as-a-service’ using Internet technology. The Cloud industry adopted three basic types of computing service models based on software level abstraction: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). Infrastructure-as-a-Service allows customers to outsource fundamental computing resources such as servers, networking, storage, as well as services where the provider owns and manages the entire infrastructure. This allows customers to only pay for the resources they consume. In a fast-growing IaaS market with multiple cloud platforms offering IaaS services, the user's decision on the selection of the best IaaS platform is quite challenging. Therefore, it is very important for organizations to evaluate and compare the performance of different IaaS cloud platforms in order to minimize cost and maximize performance.

Using a vendor-neutral approach, this research focused on four of the top IaaS cloud platforms- Amazon EC2, Microsoft Azure, Google Compute Engine, and Rackspace cloud services. This research compared the performance of IaaS cloud platforms using system-level parameters including server, file I/O, and network. System-level benchmarking provides an objective comparison of the IaaS cloud platforms from performance perspective. Unixbench, Dbench, and Iperf are the system-level benchmarks

chosen to test the performance of the server, file I/O, and network respectively. In order to capture the performance variability, the benchmark tests were performed at different time periods on weekdays and weekends. Each IaaS platform's performance was also tested using various parameters. The benchmark tests conducted on different virtual machine (VM) configurations should help cloud users select the best IaaS platform for their needs. Also, based on their applications' requirements, cloud users should get a clearer picture of which VM configuration they should choose. In addition to the performance evaluation, the price-per-performance value of all the IaaS cloud platforms was also examined.

## Chapter 1

### INTRODUCTION

Cloud computing enables on-demand network access to the collective pool of configurable physical resources such as servers, storage, and networks. These resources can then be easily managed and accessed at a minimal cost [Mell11]. Moreover, resource availability and the economic advantages of cloud computing have changed the way IT services are implemented and delivered to users and organizations [Rossa14]. Cloud computing is based on a two-tier technology: deployment models and delivery services. The three basic cloud deployment models are private cloud, public cloud, and hybrid cloud. Cloud delivery services are primarily classified as Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS) [Shawish14].

IaaS, the lowest tier in the service model stack, provides a standard virtualized operating environment to its users [Tata11]. In the IaaS model, the cloud providers own and manage the physical resources such as servers, storage, networking, and virtualization. The IaaS users rent the computing resources from the cloud providers and are responsible for managing applications, data, run time, middleware, and operating systems on self-service virtual machines. IaaS cloud services can be offered as public IaaS cloud, private IaaS cloud, and hybrid IaaS cloud [Tata11]. The public IaaS model has the potential advantages of rapid implementation, massive scalability, and consistent availability of

resources that are billed on a usage basis [Combs12]. Small and medium-size businesses (SMBs) and startup companies benefit from IaaS cloud adoption due to their business agility and low infrastructure costs [Posey15]. This study focuses on system-level evaluation of public IaaS cloud platforms' offerings.

## 1.1 Standardization and Transparency

Organizations need to comprehensively assess the technical and business requirements of an application before deploying it to the cloud environment [CloudSpectator15]. Once the technical and business requirements are assessed, the next step is to choose the right IaaS platform. [Lavnevich16].

The decision regarding the selection of an IaaS platform is crucial to an organization's success, as the efficiency and productivity of the deployed application depends on the cloud provider's capabilities [CloudSpectator15]. With the substantial increase in the number of public IaaS cloud platforms over the past decade, the selection of an appropriate IaaS platform could take great deal of time and effort. Moreover, IaaS cloud platforms have diverse service portfolios and pricing structures [Profitbricks13]. In addition, there are no uniform cloud management standards across the different IaaS cloud platforms. The lack of standardization in the IaaS industry results in the degradation of quality of services, compliance issues, service outages, interoperability issues, hidden costs, and performance bottlenecks [Ortiz11].

In the IaaS market, uncertainty with existing IaaS cloud platforms, as well as the rapid emergence of new IaaS cloud platforms into the market, leads to intense competition [CloudSpectator15]. To withstand the competition and prevent users from migrating to other IaaS cloud platforms, the chosen IaaS platform will often hide certain details from their users [Lavnevich16]. This lack of transparency is another major setback in the IaaS industry [Pwc11]. An unknown underlying infrastructure, different pricing structures, and organizational challenges are factors that make it difficult to select the right IaaS platform. An inappropriate selection of an IaaS cloud platform could lead to potential application performance issues and unnecessary IT costs [Tata11]. This study follows a data-driven approach from both the performance and price standpoints to achieve a standardized comparison of IaaS services.

## 1.2 Performance Evaluation and Benchmarking

Most of the cloud deployment decisions are based on factors such as Service Level Agreements (SLAs), datacenter location, and security, which can affect the efficiency of an application [Perry10]. However, performance is another key determinant factor in the cloud deployment decision since it significantly impacts the annual operational costs and quality of services which is sometimes overlooked due to its complexity [CloudSpectator15]. Therefore, the performance capabilities of IaaS cloud platforms must be evaluated and compared using standard measures in order for them to be useful. This can be achieved by benchmarking the performance characteristics of IaaS cloud



platforms at the system-level to establish baseline performance expectations [CloudHarmony14].

Benchmarking provides an objective comparison of cloud performance across different IaaS cloud platforms [Wise11]. Benchmarking allows users to identify the performance capabilities of each IaaS cloud platform, thereby avoiding any impact on the production environment and eliminating the post-deployment rework and unnecessary costs [CloudSpectator15]. Benchmarking is an unbiased approach that has the potential of bringing IaaS cloud providers and buyers together [Wise11]. IaaS cloud platforms benefit from this approach by identifying their own underperforming services, allowing them to make targeted improvements [Pwc11].

System-level benchmarking is used to evaluate the performance of system parameters that affect the overall application performance [CloudSpectator15]. In a multi-user cloud environment, the system resources are shared, impacting the performance of an application [Makroo16]. System-level benchmarks provide a standard method of measuring performance via usage of system-level parameters such as CPU, RAM, storage, and I/O. Standard system-level benchmarks- Unixbench [Github17], Dbench [Dbench17], and Iperf [Iperf17] are used to test the performance of server, file I/O, and network, respectively. These system-level benchmarks allow users to tune various parameters for testing the respective performance metrics under different scenarios. They also give a clear understanding of how the virtual machines running different applications affect the performance in real time [CloudSpectator15].

### 1.3 Cloud Architectures

The four leading public IaaS cloud platforms- Amazon EC2, Microsoft Azure services, Google Compute Engine, and Rackspace cloud services have been chosen for comprehensive performance evaluation. The selection of these IaaS cloud platforms is based on their reliability, flexibility, ease of use, as well as their popularity in the cloud market.

#### 1.3.1 Amazon EC2

Amazon Web Service (AWS) is the pioneer in the IaaS cloud market and is known for its agile IaaS product portfolio. Amazon Elastic Compute (EC2) is one of the services offered by AWS. Amazon EC2 enables customers and organizations of all sizes to utilize the services according to their application needs [Azavea14]. Customers are billed for each hour of resource consumption. Amazon Web Services (AWS) uses Xen virtualization technology [Posey15].

Amazon EC2 instances differ by the amount of computing resources, including processing power, memory, storage, and network connectivity. Amazon EC2 supports a wide range of operating systems, including Linux and Windows Server [Posey15].

Amazon EC2 distributes the instance families across six categories based on instance configuration [Azavea14]. Table 1 provides the description of instance types offered by Amazon EC2.

Instance Families	Description	Use cases
General purpose	These instances have burstable performance CPUs.	Web applications and development environments
Compute-optimized	These instances are designed to deliver highest performing processors.	Web servers, batch processing, and compute-intensive HPC applications
Memory-optimized	These instances are designed to deliver high memory.	Distributed memory caches and analytics, high performance databases
Storage-optimized	These instances provide fast SSD-backed instance storage.	Hadoop, data warehousing applications, and NoSQL databases
Graphics Processing Unit (GPU)	These instances are designed for graphic-intensive applications.	High performance 3D applications and video editing

Table 1: Description of Amazon EC2 Instance Types

### 1.3.2 Google Compute Engine

Google Compute Engine (GCE) is another top IaaS platform in the IaaS market. GCE is the infrastructure component of the Google cloud platform. GCE resources are hosted in data centers located in three regions across the world: US, Europe, and Asia-Pacific [Lavnevich16]. The hypervisor opted by Google is an open source Linux Kernel-based Virtual Machine (KVM). GCE supports multiple Linux OS versions including Debian, CoreOS, CentOS, OpenSUSE Enterprise, Red Hat Enterprise Linux (RHEL), and Ubuntu, as well as Windows Server [Posey15].

GCE charges customers in minute-level increments for pay-as-you-go usage [GCE17A]. Customers are initially charged for ten minutes of usage, followed by one minute increments. GCE's interconnect feature enables direct connectivity to the cloud via

Virtual Private Network (VPN) [Lavnevich16]. GCE allows its users to integrate their applications with other Google services such as Gmail, Search, and Maps.

In addition to the pre-packaged machine types, GCE also offers custom machine types. Custom machine types enable users to select any number of virtual CPUs (vCPUs) and amount of memory for their workloads [GCE17A]. Customized server configuration helps application developers select and pay for the required amount of compute and memory resources instead of paying for underutilized resources [Profitbricks13]. GCE's pre-packaged machine types are categorized into four types that offer different configurations based on compute, memory, and storage [GCE17A]. Table 2 shows the description of pre-packaged machine types offered by Google Compute Engine (GCE).

Machine types	Description	Use cases
Standard	These virtual machines have a balance of CPU and memory needs.	Web applications, application servers, database servers, and backend tasks.
High-CPU	These virtual machines offer more virtual cores relative to memory.	CPU-intensive applications
High-memory	These virtual machines offer more memory relative to virtual cores	Memory-intensive applications
Shared-core	These virtual machines offer bursting capabilities.	Small non-resource intensive applications.

Table 2: Description of Google Compute Engine Machine Types

### 1.3.3 Microsoft Azure

Microsoft Azure is another fast-growing IaaS platform. Microsoft public cloud is built on Windows Server and Hyper-V. Microsoft Azure virtual machines are charged on a

minute of usage basis [Nottingham16]. Microsoft IaaS services are available in multiple datacenters located worldwide including the United States, China, Japan, Singapore, the Netherlands, and Hong Kong [Lavnevich16]. Azure IaaS supports multiple Linux OS versions including Ubuntu Server, CoreOS, OpenSUSE, CentOS, and SLES, Debian, and Red Hat Enterprise Linux, as well as Windows Server [Posey15].

Virtual machines in Azure are categorized into Basic-tier and Standard-tier instances. Basic-tier instances do not require load balancing and auto scaling; however, Standard-tier instances include load balancing and auto scaling at no additional cost [Kwa14]. Standard-tier instances provide an optimal set of compute, memory, and I/O resources [Nottingham16]. Basic-tier is available only for general purpose instances where as Standard-tier is available across memory-intensive, and compute-intensive instances, GPU instances, and HPC instances. Table 3 provides the description of virtual machines offered by Microsoft Azure.

VM categories	Description	Use cases
General purpose	These instances have a balanced CPU to memory ratio.	Small to medium databases and low to medium traffic web servers.
Compute-optimized	These instances have a high memory to CPU ratio	Medium traffic web servers, network appliances, batch processes, and application servers
Memory-optimized	These instances have a high memory to core ratio.	Relational database servers, medium to large caches, and memory analytics
Graphic Processing Unit (GPU)	These are the specialized VMs with GPU capabilities	Graphic intensive workloads with heavy graphic rendering and video editing
High Performance Compute (HPC)	These instances offer the most powerful VMs with optimal high-throughput network interfaces	HPC workloads such as Financial risk modeling

Table 3: Description of Microsoft Azure Virtual Machines

#### 1.3.4 Rackspace Cloud Services

Rackspace OpenStack cloud computing platform is an IaaS platform that has evolved from a managed hosting service and collocation provider [Rackspace16]. The pay-as-you-go pricing model offered by Rackspace services includes a combination of infrastructure level and service level [Azavea14]. Their IaaS infrastructure is offered with two managed service levels including Managed Infrastructure and Managed Operations. These two service levels help customers easily manage and operate their own infrastructure [Rackspace16].

Rackspace has six global data centers located in Chicago, Dallas, Northern Virginia, London, and Hong Kong [Lavnevich16]. Rackspace supports different operating systems which include CentOS, CoreOS, Debian, Fedora, FreeBSD, Gentoo, openSUSE, RHEL, and Ubuntu, as well as Windows Server images - Windows Server 2008 and Windows 2012 [Posey15]. Rackspace servers are categorized as 'Flavors' based on the amount of vCPU, disk I/O, and RAM [Rackspace16]. Table 4 provides the description of flavors offered by Rackspace cloud services.

Flavors	Description	Use cases
General purpose	vCPUs of these servers are oversubscribed and burstable.	Low to medium traffic web servers, batch processing tasks, and small to medium databases
Compute-optimized	These servers have high vCPU allocation.	Web server, application servers, CPU intensive workloads, and network appliances
Memory-optimized	These servers have the larger allocations of low latency RAMs.	Medium to large caches, in-memory analytics, and search indexes
I/O-optimized	These servers have separate system and data disks for fast and sustained disk access.	Cassandra and MongoDB

Table 4: Description of Rackspace Cloud Servers

#### 1.4 Research Objectives

This study evaluates and compares the performance and price of the services provided by the four leading IaaS cloud platforms. The metrics used to analyze the performance of IaaS cloud platforms are server performance, file I/O throughput, and network

throughput. System-level benchmarks used are Unixbench, Dbench, and Iperf. The series of benchmark tests are run on three virtual instances categorized as general purpose instance, compute-intensive instance, and memory-intensive instance. The benchmark tests on these three virtual instance categories can help cloud users identify which IaaS cloud platform is the best fit for their application requirements [CloudSpectator15]. The variability in performance of all the IaaS cloud platforms is determined by running the three benchmark tests at peak and non-peak hours on different days. In addition to the performance assessment, the price-per-performance analysis is calculated to determine which IaaS cloud platform delivers the best performance at minimum cost.



## Chapter 2

### LITERATURE REVIEW

#### 2.1 Studies on Performance and Performance Variability

Leitner and Cito conducted a constructive analysis of performance variability in the services offered by public IaaS cloud platforms [Leitner16]. The detailed hypotheses were formulated on the performance variation and predictability in cloud platforms offering IaaS services. The formulated hypotheses were validated and analyzed by executing the micro and application-level benchmarks on four IaaS cloud platforms: Amazon EC2, Google Compute Engine, Microsoft Azure, and IBM Softlayer. Three micro benchmarks were used to test CPU speed, disk I/O, and memory performance and two application-level benchmarks were run to test OLTP and Java compilation. The benchmark tests were performed on three different VM configurations (micro, small, and medium instance types) launched in different geographical regions. The series of benchmark tests were executed six times a day over a period of one month to test performance variability.

The results indicated that hardware heterogeneity and multi-tenancy have an impact on performance and performance variability on few IaaS cloud platforms [Leitner16]. The series of benchmark tests indicated that the performance of CPU-bound applications

varies primarily due to hardware heterogeneity whereas the performance of I/O bound applications varies due to noisy neighbor effect. The performance and predictability of a cloud instance depends significantly on time of the day and day of the week as well as selected geographical region.

Schad et al. carried out a study on performance variability of two IaaS cloud platforms- Amazon EC2 and Rackspace cloud services [Schad10]. Ubench benchmark was used to measure CPU and memory speed. Bonnie and Iperf benchmarks were used to test performance of disk I/O and network respectively. Along with these benchmarks, MapReduce applications were run on the IaaS cloud platforms' infrastructure to test their application performance. Performance variability was analyzed by running benchmark tests on small and large instances over a period of one month, in different availability zones.

The authors concluded from the results that geographical location has an impact on performance variability [Schad10]. The results indicated that CPU, I/O, and network performance of small and large instances were very unpredictable. The reason for such variability was observed due to different system types used by virtual modes i.e. the performance of Xeon-based systems was better when compared to that of Opteron-based systems.

Lenk et al. proposed a new benchmarking method that determines the actual performance of the virtual machines running a specific IaaS service [Lenk11]. The proposed

benchmarking method was tested on different virtual instances of three IaaS cloud platforms: Amazon EC2, Flexiscale, and Rackspace cloud. Benchmarks from the Phoronix test suite were used to build their customized benchmark suite [Lenk11]. The paper concluded that the performance specifications provided by the IaaS cloud platforms were inadequate to predict the actual performance of the deployment. The benchmark suite did not test the network performance. Also, the paper did not consider a performance-cost analysis when comparing the IaaS cloud platforms.

Major findings of the benchmark results indicated that there is a strong relation between the performance of a VM and its underlying CPU architecture [Lenk11]. The results obtained from benchmark suite showed that data center location, benchmark start and execution times have no strong impact on the performance. Furthermore, the results indicated that time related dimensions do not have major impact on the performance of a VM.

## 2.2 Studies on Pricing

Kihal et al. address the lack of transparency in the pricing structure of IaaS cloud platforms [Kihal12]. Their research aimed to analyze the pricing structure of IaaS cloud platforms by employing two different price comparison methods. The pricing study evaluated five IaaS cloud platforms - Amazon EC2, IBM Cloud, Microsoft Azure, Terremark IaaS platform, and Google Compute Engine. The first method was "hedonic pricing" which decomposes IaaS prices into characteristics of the goods being sold and

external factors affecting pricing. Another method was "PriCo" which considers the prices of competitive platforms. The two pricing comparison methods could help customers identify the least expensive provider for their applications and help IaaS cloud platforms self-assess their pricing position in the market.

Hedonic pricing method indicated that Amazon EC2 and Microsoft Azure are more suitable for high computing requirements, whereas Google, IBM, and Terremark are more suitable for lower computing requirements. PriCo method indicated that Google is the most suitable platform for applications with low-CPU and high-memory requirements and Terremark is the most favorable provider for applications with high-CPU and low-memory requirements. Both pricing comparison methods indicated that Google is the least expensive provider for most of the computing requirements [Kihal12].

Although numerous studies exist in the literature, none provided a comprehensive evaluation of IaaS cloud platforms from both performance and price standpoints under different scenarios. The existing research provides only low-level performance evaluation of IaaS cloud platforms. Currently, there are no research methodologies available that can provide a clear analysis to identify the best IaaS cloud platform for specific application requirements. In this study, the performance capabilities of IaaS cloud platform were extensively evaluated by performing stress testing on the server, file system, and network to assess how the IaaS cloud platforms perform under unexpected, extreme loads.

## Chapter 3

### RESEARCH METHODOLOGY

This chapter establishes benchmarking criteria of the server performance, file I/O performance, network performance, and the price-per-performance analysis.

#### 3.1 Server Performance

In cloud environments, every cloud provider has their own way of virtualizing the CPU cores [CloudSpectator15]. The virtual machines (VMs) do not run in isolation [Azavea14]. Rather, they run on a physical host along with a variety of other virtual machines provisioned to other users. The overall load on the host will affect the performance of any individual VM configured on that host [Azavea14]. The virtual machine that affects the performance of other virtual machines of the same physical host is called a noisy neighbor tenant [Makroo16]. The noisy neighbor tenant monopolizes the host resources, including CPU cores, resulting in performance fluctuations across all the instances of the host [Kajeeepeta10].

The performance of a virtual machine depends on many factors such as CPU speed, operating system, processor type, internal memory, and underlying virtualization technology [CloudHarmony14]. There are other factors that affect performance that are

out of user's control, such as state of services, noisy neighbors, and hardware heterogeneity [Makroo16]. A generalized system benchmarking tool can be leveraged to compare the system performance of identically configured VMs of different vendors. As mentioned earlier, the overall performance of a virtual machine depends on the number of virtual CPU cores, thus it will be useful to accumulate metrics on a per-core basis. That is, the benchmarking tool has to be executed in both single-core and multi-core modes. These tests are useful in determining whether multiple cores are fully utilized while performing certain tasks [Github17].

In multi-tenant environments with sophisticated virtualization techniques for sharing the resources among users, the performance of the server may vary based on availability of CPU resources due to varying loads on the host in a given day [Persico10]. Therefore, the benchmarking exercise needs to be performed at peak hours and non-peak hours of weekdays and weekends.

### 3.2 File I/O Performance

Cloud platforms offer different cloud storage technologies ranging from traditional hard drives (HDD) to Solid State Drives (SSD), with or without RAID support [Wells15]. Each storage option delivers different levels of performance, redundancy, and availability [Profitbricks13]. The storage performance also depends on whether the storage is temporary or persistent. As a result, the efficiency or the speed at which the data is transferred internally is a key factor that varies among the platforms [Keating16]. Storage

performance is evaluated by generating customized I/O server workloads to test its efficiency when handling the workload [Dbench17]. The server can be stressed by emulating multiple client processes to determine how well it can handle file requests from concurrent clients. The file I/O capabilities can be further tested by carrying out the tests under systematically defined load conditions to determine how the server performs.

Performance variability also needs to be evaluated since performance may vary at different times due to multi-tenancy [CloudSpectator15]. One or more VMs on the same physical host running heavy application workloads such as I/O-intensive applications may consume most of the disk I/O [Makroo16]. As a result, other VMs residing on the same physical host may suffer from resource contention.

### 3.3 Network Performance

Networking is a key factor in cloud computing since the LAN capability in the cloud is virtualized. In a cloud environment, the data stored and processed in virtual environments are distributed across multiple physical servers [Moghul12]. The networking capabilities, such as network infrastructure, load balancing technology, as well as network resource allocation strategies, are different among the platforms due to technological differences [Persico15]. Furthermore, many IaaS cloud platforms use 10 Gbps interconnect with maximum throughput limited up to 1 Gbps, based on the size of VM [Liu16]. Also, the IaaS cloud platforms advertise maximum allocated bandwidth for different VM configurations [AmazonEC217B]. Therefore, it is important to examine how each IaaS

platform actually delivers the network throughput in comparison with its theoretical/advertised bandwidth.

The network can be stressed by allowing parallel data streams from client to server. Stressing the network allows the user to fully test the network capacity of the provider [Iperf17]. Sometimes, heavy network traffic consumes more CPU time, affecting the network throughput and latency [Mytton14].

The variability in network performance occurs when the majority of available network bandwidth is utilized by other hosts on the same physical infrastructure [Azavea14]. The time of day and day of the week also contribute to the variability in network performance. It is assumed that the network delivers lower throughput during the peak hours, or when backups are running [Cook08]. The variation in network performance causes difficulties for the applications that are performing latency-sensitive tasks [Posey15]. The applications running on other VMs of the same physical host may cause variation in network performance of a particular VM [Persico15]. Therefore, it is necessary to test the network performance during peak hours and non-peak hours of weekdays and weekends.

### 3.4 Performance Variability

In a public cloud environment, the performance of an application is not always consistent [Persico15]. A series of factors including workload variability, virtualization overhead, and resource-time sharing can contribute to performance variability [Jackson11].



The variability in performance needs to be determined using the benchmark results of different IaaS cloud platforms obtained across different test sessions. There are certain techniques or measures to emphasize the performance variability of cloud platforms for all the three system-level tests. The common statistical functions that measure the variability in performance are variance, mean, and standard deviation [Schad10]. These statistical functions are used to derive the Coefficient of Variation (CoV). The Coefficient of Variation can also be used as a simple measure of relative variability of performance [Schad10].

### 3.5 Pricing

Pricing is an important factor that needs to be considered for IaaS platform selection by business organizations. The value of a IaaS offering is vital for the organizations of any scale when determining the amount of resources needed to deploy an application into a provider's environment [Profitbrick13]. Organizations can achieve better business outcome when an application deployed on the cloud generates higher performance in a cost-effective manner [Perry10]. The combination of price and performance can be used to identify the platform that offers best IaaS services at minimal cost [Profitbricks13].

## Chapter 4

### METRICS AND SPECIFICATIONS

#### 4.1 Metrics

The following metrics are examined when comparing the IaaS cloud platforms:

1. Unixbench Score: The scores of individual tests - Dhrystone, Whetstone, Execl throughput, File Copy, Pipe throughput, Pipe-based Context Switching, Process Creation, Shell Scripts, System call Overhead, Graphical tests are aggregated to an overall index score for the system [Github17]. A higher score represents better server performance. The following is the brief description of individual tests:

- Dhrystone: Dhrystone test is used to measure the efficiency of string handling functions.
- Whetstone: Whetstone is used to measure the speed and efficiency of integer and floating-point arithmetic operations.
- Execl Throughput: This test is used to measure the number of execl calls that can be performed per second.
- File Copy: File Copy test is used to measure the rate at which the data can be transferred from one file to another, using various buffer sizes.
- Pipe Throughput: This test is used to measure the rate at which a process

can write 512 bytes to a pipe and read them back (communication between processes).

- Pipe-based Context Switching: This test is used to measure the rate at which two processes can exchange an increasing integer through a pipe.
  - Process Creation: This benchmark test is used to measure the number of times a process can fork another process.
  - Shell scripts: This benchmark test is used to measure the rate at which a process can start and reap a set of one, four, and eight concurrent copies of shell scripts.
  - System call Overhead: This test is used to measure estimated cost of entering and exiting the operating system kernel.
  - Graphical tests: Graphical tests are used to measure the system's 2D and 3D graphics performance.
2. File I/O throughput: This throughput is defined as the rate at which the data is moved to or from storage. File I/O throughput is expressed in Megabytes per second [Dbench17].
  3. Internal network throughput: This throughput is the rate at which data are transmitted over the network between the test VMs. Network throughput is measured in bits per second [Iperf17].
  4. Price-per-performance value: Price-per-performance is the ratio of the benchmark's performance to the hourly VM charges of the provider [Profitbricks13].

## 4.2 Hardware and Software Specifications

### 4.2.1 Hardware Specifications

Unixbench, Dbench, and Iperf benchmark tools are installed, configured, and executed on Linux VMs connected using SSH Client. The pre-configured virtual machines offered by Amazon EC2, Microsoft Azure, and Rackspace cloud services make it difficult to match the exact target server configurations [CloudSpectator15]. To ensure meaningful comparisons, the number of vCPUs and RAM are matched across the IaaS cloud platforms. Instances or virtual machines are divided into different types for comparable use cases: general purpose instances, compute-intensive instances, and memory-intensive instances. Tables 5, 6, and 7 display the VM parameters of four IaaS cloud platforms under the three categories of instance types.

IaaS Cloud Platforms	Offering name	Number of vCPUs	RAM(GB)	Advertised Network Bandwidth
Amazon EC2	t2.micro	1	1GB	Less than 500Mbps
Google Compute Engine	Customized	1	1GB	2Gbps
Microsoft Azure	Standard-A1	1	1.75GB	500Mbps
Rackspace	General1-1	1	1GB	200Mbps

Table 5: General Purpose Instances across Cloud Providers

IaaS Cloud Platforms	Offering Name	Number of vCPUs	RAM(GB)	Advertised Network Bandwidth
Amazon EC2	r3.xlarge	4	30.5GB	500Mbps
Google Compute Engine	n1-highmem4	4	26GB	8Gbps
Microsoft Azure	Standard D12-v2	4	28GB	3Gbps
Rackspace	Memory1-30	4	30GB	1.3Gbps

Table 6: Memory-Intensive Instances across Cloud Providers

IaaS Cloud Platforms	Offering Name	Number of vCPUs	RAM(GB)	Advertised Network Bandwidth
Amazon EC2	c4.2xlarge	8	15GB	1000Mbps
Google Compute Engine	Customized	8	15GB	16Gbps
Microsoft Azure	Standard F8	8	16GB	5Gbps
Rackspace	Compute1-15	8	15GB	1.3Gbps

Table 7: Compute-Intensive Instances across Cloud Providers

#### 4.2.2 Software Specifications

Unixbench 5.1.3, Dbench, and Iperf 3.0 benchmark tools are installed and configured on all the three instances. Additional packages, such as make, GCC, automake, libpopt-dev, and g++ were installed to compile the benchmarks.

All the instances across the four IaaS cloud platforms were instantiated on the 64-bit Red Hat Enterprise Linux operating system. The SSH client, PuTTY, was used to connect to the Linux VMs on which the benchmark tools were installed and executed. PuTTYgen was employed to convert the private key into .ppk format. WinSCP was used to transfer files between the virtual machine and local machine where benchmarks were executed.

## Chapter 5

### TEST METHODOLOGY

UnixBench, Dbench, and Iperf benchmarks are used to evaluate the server performance, file I/O performance, and internal network performance, respectively. This section describes how the benchmark tests were executed.

There are certain key conditions that need to be considered before conducting benchmark tests. While running any benchmark on a specific instance, it is important to make sure that no other applications are running on that instance, as other applications would consume system resources resulting in inaccurate results. Every test session was performed on a newly instantiated VM allowing for the possibility of switching physical server throughout the test period, which would enable accurate identification of performance variability. All the instances tested across the IaaS cloud platforms were launched in US data centers to ensure uniform comparisons. To test the internal network performance using Iperf benchmark, two virtual instances of the same VM configuration were instantiated where one instance acts as server and the other as client.

- Performance benchmarking and performance variability

IaaS cloud platforms were tested and evaluated by conducting the benchmarks to evaluate server performance, file I/O performance, and network performance.

Performance variability is determined by statistically analyzing the benchmark results

over the time [Schad10]. The benchmark tests were run at peak hours and non-peak hours, on business days and on the weekend for a total of four sessions, with each session representing the test occurrences for the week.

- Price-per-performance analysis

Price-per-performance analysis is determined to identify the best-valued IaaS platform. It is calculated by dividing the IaaS cloud platforms' benchmark result by hourly VM charges.

## 5.1 Performance Benchmarking and Performance Variability

### 5.1.1 Server Performance

UnixBench is the system benchmark suite used to provide a basic indicator of the system performance of a Unix-like system [Github17]. UnixBench consists of number of individual tests - Dhrystone, Whetstone, Execl throughput, File Copy, Pipe throughput, Pipe-based Context Switching, Process Creation, Shell Scripts, System call overhead, and Graphical tests. Each of these tests evaluates various aspects of the system's performance. A single run of the benchmark generates individual index values of all the tests and produces an overall index of the system [Github17]. The testing methodology is as follows:

1. Create and launch a general purpose instance on all four IaaS cloud platforms.

2. Install and configure Unixbench on each general purpose instance of all four IaaS cloud platforms (See Appendix A for details on Unixbench installation and setup).
3. Run the benchmark for Single-threaded tests and Multi-threaded tests by specifying the number of CPU cores to use and number of parallel tests.  
  
Single-threaded tests use one CPU core and run a single copy of test program at a time, whereas Multi-threaded tests utilize multiple CPU cores and run multiple copies of the same tests.
4. Run each test twice per session and then calculate the average score for each test.
5. Calculate the median for the set of respective average scores captured from different sessions for both Single-threaded and Multi-threaded tests.
6. Compare the median scores across the IaaS cloud platforms.
7. For both Single-threaded and Multi-threaded tests, determine performance variability for the aggregate scores across sessions on all the IaaS cloud platforms.
8. Analyze the performance variability using T-test function and Coefficient of Variation %. Calculate P-values for each pair of providers on the aggregated scores captured from each test session.
9. Follow the steps (1) through (8) on memory-intensive instances across all the IaaS cloud platforms.
10. Follow the steps (1) through (8) on compute-intensive instances across all the IaaS cloud platforms.



### 5.1.2 File I/O Performance

Dbench benchmark generates I/O workloads to file system [Dbench17]. This benchmark only performs file system calls, but does not perform networking calls. The purpose of Dbench is to predict how the server can handle concurrent client processes performing the workload without any lag in the response time. All the throughput values of individual client processes are aggregated to the average throughput of the server.

Dbench benchmark was executed to perform two different tests- standard file I/O tests and file I/O performance overhead test. Standard file I/O test is performed by executing the benchmark with standard file I/O options by varying the number of client processes. File I/O performance overhead test is performed to by executing the synchronous file operations by varying the number of clients. File I/O performance overhead tests determine how the server can handle under unexpected load conditions. The synchronous file operations create extra writes to the disk resulting in the performance overhead on the server (See Appendix B for Dbench installation and setup). The testing methodology is as follows:

1. Create and launch a general purpose instance on all four IaaS cloud platforms.
2. Install and configure Dbench on each general purpose instance of four IaaS cloud platforms (See Appendix B for details on Dbench installation and setup).
3. On each test session, run the benchmark to test standard file I/O throughput and file I/O overhead throughput, for each of the four IaaS cloud platforms.

4. Configure the benchmark parameters by varying the number of client processes (1, 2, 25, 50, 100, 200, and 300) and capture the throughput results.
5. For each IaaS platform, calculate the median for the set of throughput values captured from all the test sessions.
6. For file I/O performance overhead, run the benchmark by tuning the parameters, including number of client processes and file I/O options.
7. For each IaaS platform, calculate the median for the set of throughput values captured from all the test sessions.
8. Compare the IaaS cloud platforms with respect to standard file I/O throughput and file I/O overhead throughput.
9. For both tests, determine the performance variability by calculating the median for the set of respective throughput results obtained from each test session.
10. Performance variability is analyzed using T-test function and Coefficient of Variation.
11. For standard file I/O test and file I/O performance overhead tests, calculate P-values for each pair of IaaS cloud platforms for the median throughput values with different set of client parameters (50, 100, 200, and 300) captured from four test sessions.
12. Repeat steps (1) through (11) for memory-intensive instances across all the IaaS cloud platforms.
13. Repeat steps (1) through (11) for compute-intensive instances across all the IaaS cloud platforms.

### 5.1.3 Network Performance

The Iperf tool measures the throughput of the network carrying TCP data streams. The purpose of the benchmark is to measure the TCP network throughput between two virtual machines [Iperf17]. The benchmark tests transfer as much data as possible through the internal network for 1800 seconds over default TCP port 5201. The tests are run with increasing the number of parallel clients to measure the QoS functionality [Mytton14]. The maximum number of clients allowed by Iperf is 128. The testing methodology is as follows:

1. Create and launch two general purpose instances on all four IaaS cloud platforms. Name the instances - Client or Server.
2. Install and configure Iperf tool on the two general purpose instances of each cloud platform (See Appendix C for details on Iperf installation and setup).
3. Enable the Server instance to listen to port 5201.
4. Allow Client instance to connect to the Server instance over the default port, using the Server's IP address.
5. Run each test iteration for 1800 seconds by varying the number of simultaneous connections (1, 50, and 128) from the Client instance to the Server instance.
6. Calculate the overall median throughput for the set of throughput results captured from four test sessions.
7. Compute the median for the set of throughput values obtained from each test session. This is to determine the performance variability.

8. Performance variability is analyzed using T-test function and Coefficient of Variation.
9. Calculate the P-values for each pair of IaaS cloud platforms for the median throughput values with different set of client parameters (50 and 128) captured from four test sessions.
10. Repeat steps (1) through (9) for memory-intensive instances across all the IaaS cloud platforms.
11. Repeat steps (1) through (9) for compute-intensive instances across all the IaaS cloud platforms.

## 5.2 Price-per-Performance Analysis

The price-per-performance is calculated on all the IaaS cloud platforms across three different instances with respect to server performance, file I/O performance, and network performance.

### 5.2.1 Price-per-Server Performance

1. On general purpose instances, price-per-server performance value for each IaaS platform is calculated by dividing Unixbench score by VM charge per hour.
2. Compare the IaaS cloud platforms against price-per-server performance values.
3. Identify the best-valued IaaS platform.
4. Repeat step (1) and step (3) for compute-intensive instance and memory-intensive instances.

### 5.2.2 Price-per-File I/O Performance

1. On general purpose instances, price-per-file I/O performance value for each IaaS platform is calculated by dividing provider's throughput by VM charge per hour.
2. Compare the IaaS cloud platforms against price-per-file I/O performance values.
3. Identify the best-valued IaaS platform.
5. Repeat the step (1) and step (3) for compute-intensive instance and memory-intensive instance.

### 5.2.3 Price-per-Network Performance

1. On general purpose instances, price-per-network performance value for each IaaS platform is calculated by dividing provider's throughput by VM charge per hour.
2. Compare the IaaS cloud platforms against price-per-network performance values.
3. Identify the best-valued IaaS platform.
4. Repeat the step (1) and step (3) for compute-intensive instance and memory-intensive instance.

## Chapter 6

### RESULTS AND ANALYSIS

This chapter summarizes the analysis of how the four IaaS cloud platforms performed with respect to server performance, file I/O, and network performance on each instance category. The best-valued IaaS platform for each benchmarking test was also determined.

The Microsoft Excel 2007 T-Test function was used to perform statistical analysis on the collected data in order to calculate P-values for the Unixbench score, file I/O throughput, and network throughput. Since the standard deviations for the two datasets may be different, the T-Test function was performed using the two-tailed distribution (tails = 2) and two-sample unequal variance (type = 3) options. The difference in the datasets is said to be statistically significant if obtained P-value is  $< 0.05$ . T-test was calculated on the pair-wise comparison of IaaS cloud platforms. All the data points from the four test sessions were used to calculate the T-test.

The Coefficient of Variation (CoV) was used as a measure to determine performance variability [Schad10]. The Coefficient of Variation is derived from variance, standard variance, and mean. It is expressed as percentage and is defined as the ratio of standard deviation and mean. Lower CoV indicates stability in performance throughout the test period. The Coefficient of Variation for each IaaS platform was calculated on the data points of each benchmark test captured across the test sessions.

The price-per-performance value was calculated to determine a provider's ability to deliver the expected performance for the price charged. It was used to identify the best-valued IaaS platform. A higher price-per-performance value indicates better performance at lower costs [Profitbricks13].

## 6.1 Server Performance

Unixbench benchmark suite was used to test the server performance running Single-threaded and Multi-threaded tasks. As discussed in Chapter 3, server performance depends on many factors such as processor, CPU clock speed, compiler, and operating system [CloudHarmony14]. All the instances across all the IaaS cloud platforms were run on the same OS (i.e. Red Hat Enterprise Linux 7). To maintain uniformity, the same version of gcc compiler was matched on all virtual instances.

### 6.1.1 General Purpose Instances

Table 8 shows the comparison of Single-threaded scores and Multi-threaded scores across the IaaS cloud platforms. The scores listed in the table are obtained by calculating the median for individual scores captured across the test sessions. Figure 1 shows the comparison of Single-threaded scores across the IaaS cloud platforms on general purpose instances. Figure 2 shows the comparison of Multi-threaded scores across the IaaS cloud platforms on general purpose instances.

IaaS Cloud Platforms	Single-threaded score	Multi-threaded score
Amazon EC2	1266.60	213.18
Google Compute Engine	1634.08	1625.20
Microsoft Azure	592.55	919.19
Rackspace	1394.38	1359.73

Table 8: Single- and Multi-Threaded Scores on General Purpose Instance

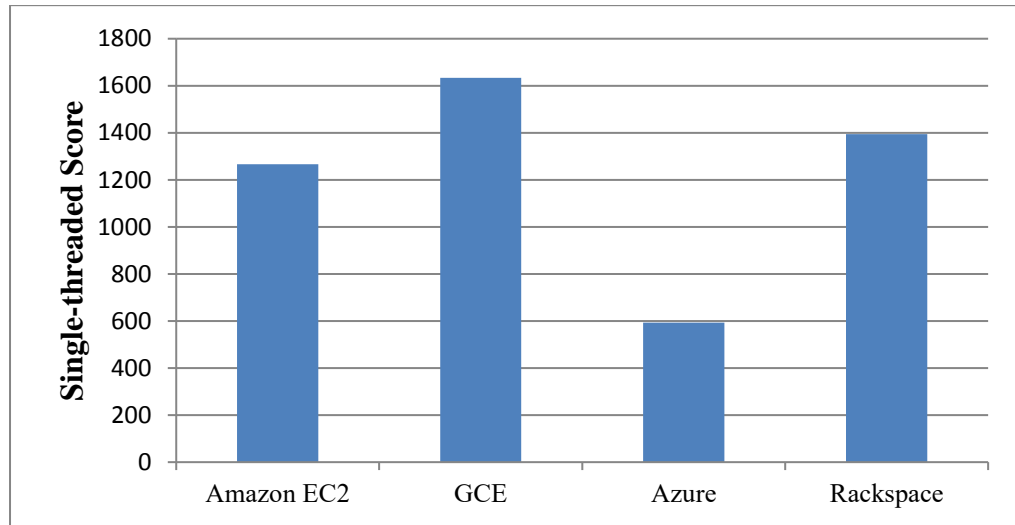


Figure 1: Single-Threaded Scores on General Purpose Instance

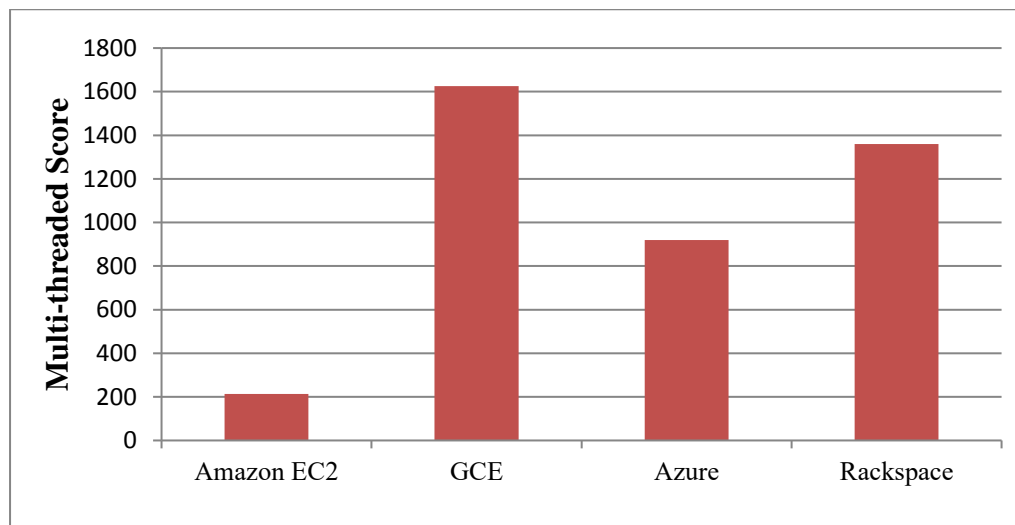


Figure 2: Multi-Threaded Scores on General Purpose Instance



GCE achieved the highest scores due to its advanced processor- Intel Xeon with a CPU speed of 2.60GHz [GCE17A]. The poorest performance of Azure in the single-threaded test was because of the inadequate physical hardware for the general purpose instance [Nottingham16]. Amazon EC2 delivered the least performance in the Multi-threaded test due to its burstable CPU core [Azavea14]. Figure 1 shows that the performance score of GCE's Single-threaded performance was 2.7 times more than that of Azure. Figure 2 indicates that the performance score of GCE was 7.6 times more than that of Amazon EC2 in the Multi-threaded test.

#### 6.1.1.1 Performance Variability

For both Single-threaded and Multi-threaded tests, P-value was calculated for each pair of IaaS cloud platforms on the scores captured from the four test sessions. Table 1 (See Appendix D for Statistical Analysis) shows the P-values for Single-threaded and Multi-threaded tests on General purpose instance. The difference in the Single-threaded scores between Amazon EC2 and Rackspace were found to be statistically insignificant with a P-value of 0.289. The differences in the Multi-threaded scores between all the pairs of cloud providers were statistically significant due to a P-value  $< 0.05$ .

Table 9 shows the Single-threaded scores of each provider throughout the test period with CoV indicating performance variability. Table 10 shows the Multi-threaded scores of each provider throughout the test period with CoV indicating performance variability. In both the tests, GCE was the most consistent IaaS platform with the least CoV. Amazon

EC2 displayed highest variability in performance, possibly due to VM resource contention between users [Makroo16].

Sessions	Amazon EC2 (score)	Google Compute Engine (score)	Microsoft Azure (score)	Rackspace (score)
Session-1	1155.60	1650.55	567.50	1489.40
Session-2	1376.45	1617.60	617.60	1522.35
Session-3	1446.05	1608.80	618.25	1279.95
Session-4	1156.75	1659.40	517.15	1299.35
Coefficient of Variation %	11.68%	1.51%	8.32%	9.00%

Table 9: Single-Threaded Scores with CoV on General Purpose Instance

Sessions	Amazon EC2 (score)	Google Compute Engine (score)	Microsoft Azure (score)	Rackspace (score)
Session-1	259.25	1639.75	598.50	1458.05
Session-2	204.10	1610.65	652.65	1494.55
Session-3	206.95	1575.45	661.45	1251.35
Session-4	219.40	1649.70	542.75	1261.40
Coefficient of Variation %	11.43%	2.06%	8.95%	9.36%

Table 10: Multi-Threaded Scores with CoV on General Purpose Instance

#### 6.1.1.2 Price-per-Server Performance on General Purpose Instance

Price-per-server performance value for each IaaS platform was determined by dividing the respective score by VM charge per hour. Table 11 shows the comparison of price-per-

server performance values across all the platforms. The price-per-server performance value was higher for GCE. The higher price-per-server performance value indicates that the IaaS platform delivers good performance at low price [Profitbricks13]. Therefore, GCE was the best-valued IaaS platform.

IaaS Cloud Platforms	Price-per-server performance value
Amazon EC2	2920.27
Google Compute Engine	56041.38
Microsoft Azure	9675.68
Rackspace	11621.62

Table 11: Price-per-Server Performance on General Purpose Instance

#### 6.1.2 Memory-Intensive Instances

Table 12 shows the comparison of Single-threaded scores and Multi-threaded scores across the IaaS cloud platforms. The scores listed in the table were obtained by calculating the median for individual scores captured across the test sessions. Figure 3 shows the comparison of Single-threaded scores across the IaaS cloud platforms on memory-intensive instance. Figure 4 shows the comparison of Multi-threaded scores across the IaaS cloud platforms on memory-intensive instance.

IaaS Cloud Platforms	Single-threaded score	Multi-threaded score
Amazon EC2	1301.53	3183.10
Google Compute Engine	1394.33	3312.48
Microsoft Azure	1473.35	4172.73
Rackspace	1301.53	3183.10

Table 12: Single- and Multi-Threaded Scores on Memory-Intensive Instance

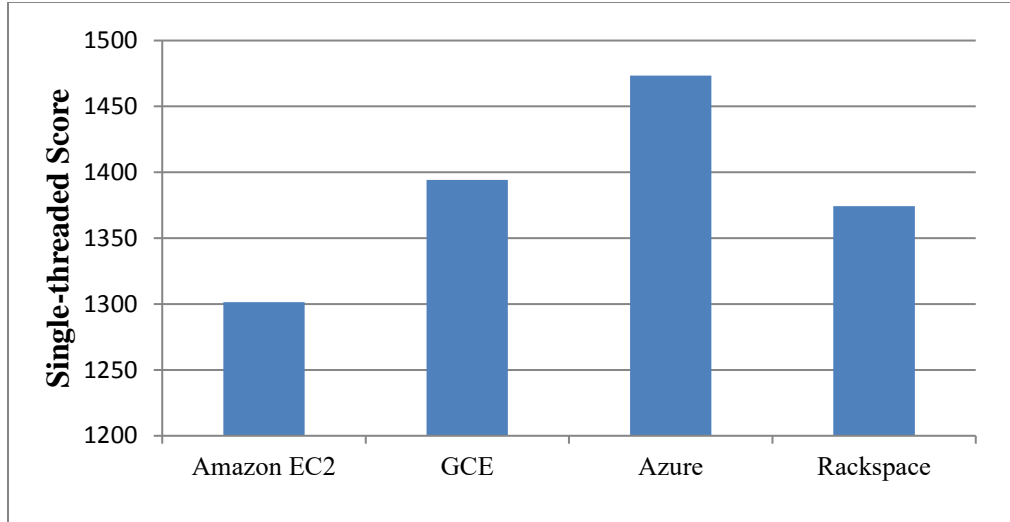


Figure 3: Single-Threaded Scores on Memory-Intensive Instance

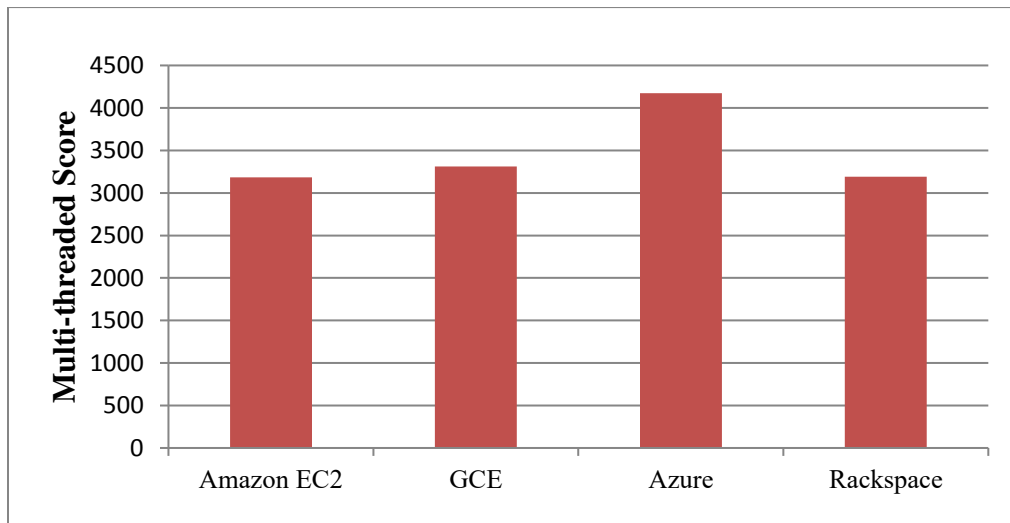


Figure 4: Multi-Threaded Scores on Memory-Intensive Instance

Figure 3 and Figure 4 indicate that Azure achieved the highest performance due to its advanced processor with increased CPU frequency of 3.2 GHz with Intel Turbo Boost Technology providing a performance boost [Nottingham16]. The lowest performance of Amazon EC2 was as result of its underlying processor-Intel Xeon with clock speed of 2.5 GHz [AmazonEC217A].

#### 6.1.2.1 Performance Variability

For both Single-threaded and Multi-threaded tests, P-value was calculated for each pair of IaaS cloud platforms on the scores captured from the four test sessions. Table 2 (See Appendix for Statistical Analysis) shows the P-values for Single-threaded and Multi-threaded tests on Memory-intensive instance. For Single-threaded and Multi-threaded tests, GCE and Amazon EC2 were statistically significant with each other with a P-value  $< 0.05$ .

Table 13 shows the Single-threaded scores of each IaaS platform throughout the test period with CoV indicating the variability in performance. Table 14 shows the Multi-threaded scores of each IaaS platform throughout the test period with CoV indicating the variability in performance. In both the tests, Amazon EC2 was the most consistent IaaS platform with the least CoV. Azure displayed the least stability in performance, possibly due to VM resource contention between users [Makroo16].

Sessions	Amazon EC2 (score)	Google Compute Engine (score)	Microsoft Azure (score)	Rackspace (score)
Session-1	1268.75	1347.30	1378.15	1435.20
Session-2	1309	1372.95	1106.10	1447.40
Session-3	1297.20	1420.35	1568.55	1313.55
Session-4	1305.85	1415.70	1615.70	1309.30
Coefficient of Variation %	1.41%	2.53%	16.33%	5.46%

Table 13: Single-Threaded Scores with CoV on Memory-Intensive Instance

Sessions	Amazon EC2 (score)	Google Compute Engine (score)	Microsoft Azure (score)	Rackspace (score)
Session-1	3109.65	3276.70	4172.20	3343.10
Session-2	3167.35	3261.95	3016.70	3380.80
Session-3	3198.85	3348.25	4173.25	3025.30
Session-4	3209.50	3404.60	4289.75	3040.65
Coefficient of Variation %	1.41%	1.99%	15.34%	5.96%

Table 14: Multi-Threaded Scores with CoV on Memory-Intensive Instance

#### 6.1.2.2 Price-per-Server Performance on Memory-Intensive Instance

Price-per-server performance value for each IaaS platform was determined by dividing the respective score by VM charge per hour. Table 15 shows the comparison of price-per-server performance values across all the IaaS cloud platforms. The price-per-server performance value was higher for GCE. Therefore, GCE was the best-valued IaaS platform.

IaaS Cloud Platforms	Price-per-server performance value
Amazon EC2	7073.56
Google Compute Engine	18714.58
Microsoft Azure	9681.51
Rackspace	5811.87

Table 15: Price-per-Server Performance on Memory-Intensive Instance

### 6.1.3 Compute-Intensive Instances

Table 16 shows the comparison of Single-threaded scores and Multi-threaded scores across the IaaS cloud platforms. The scores listed in the table were obtained by calculating the median for individual scores captured across the test sessions. Figure 5 shows the comparison of Single-threaded scores across the IaaS cloud platforms on compute-intensive instance. Figure 6 shows the comparison of Multi-threaded scores across the IaaS cloud platforms on compute-intensive instance.

IaaS Cloud Platforms	Single-threaded score	Multi-threaded score
Amazon EC2	1604.43	5819.53
Google Compute Engine	1447.75	5218.90
Microsoft Azure	1492.65	6331.30
Rackspace	1414.28	4423.18

Table 16: Single- and Multi-Threaded Scores on Compute-Intensive Instance

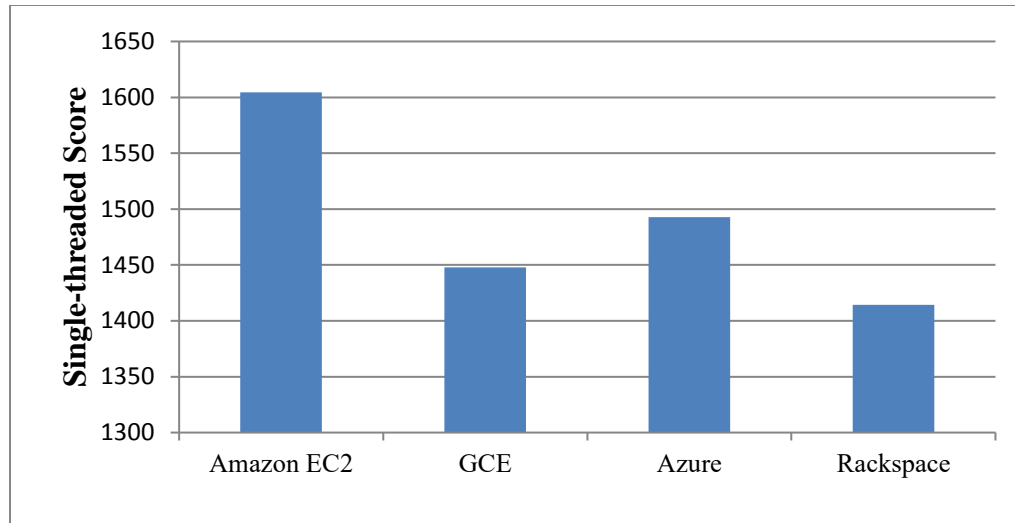


Figure 5: Single-Threaded Scores on Compute-Intensive Instance

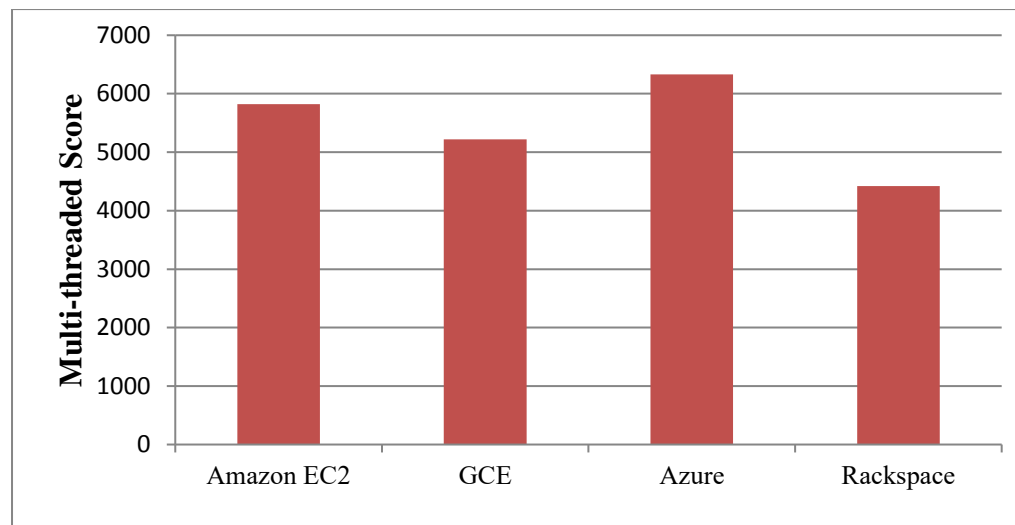


Figure 6: Multi-Threaded Scores on Compute-Intensive Instance

Figure 5 indicates that Amazon EC2 achieved the highest performance in the Single-threaded test because its compute-intensive instance runs on a customized processor at a speed of 2.9 GHz. This instance was designed to deliver the highest level of processor performance [Barr14]. Figure 6 indicates that Azure delivered highest Multi-threaded



performance because its processor with a high CPU clock speed of 3.1 GHz provides a performance boost by utilizing all eight vCPUs consistently [Nottingham16].

#### 6.1.3.1 Performance Variability

For both Single-threaded and Multi-threaded tests, P-value was calculated for each pair of IaaS cloud platforms on the scores captured from the four test sessions. Table 3 (See Appendix D for Statistical Analysis) shows the P-values for Single-threaded and Multi-threaded scores on Compute-intensive instance. The differences in Single-threaded scores by two pairs of IaaS cloud platforms - GCE Vs Amazon EC2, and Amazon EC2 Vs Rackspace were statistically significant. The differences in Multi-threaded scores by four pairs of IaaS cloud platforms - GCE Vs Amazon EC2, GCE Vs Rackspace, Amazon EC2 Vs Rackspace, and Rackspace Vs Azure were statistically significant with a P-value < 0.05.

Table 17 shows the Single-threaded scores of each IaaS platform throughout the test period with CoV indicating the variability in performance. Table 18 shows the Multi-threaded scores of each provider throughout the test period with CoV indicating the variability in performance. In both the tests, Amazon EC2 was the most consistent IaaS platform with the least CoV. Azure displayed the least consistency in performance. One possible explanation for the performance variability in Azure was due to the VM resource contention between users [Makroo16].

Sessions	Amazon EC2 (score)	Google Compute Engine (score)	Microsoft Azure (score)	Rackspace (score)
Session-1	1605.6	1438.35	1053.8	1338.75
Session-2	1603.25	1378.1	1299.55	1366.95
Session-3	1601.85	1457.15	1685.75	1467
Session-4	1617.9	1463.9	1687.15	1461.6
Coefficient of Variation %	0.46%	2.72%	21.72%	4.64%

Table 17: Single-Threaded Scores with CoV on Compute-Intensive Instance

Sessions	Amazon EC2 (score)	Google Compute Engine (score)	Microsoft Azure (score)	Rackspace (score)
Session-1	5790.2	5303.75	6454.1	4069.5
Session-2	5881.1	5009.75	4580.4	4294.35
Session-3	5772	5173	6335.2	4602.15
Session-4	5848.85	5264.8	6327.4	4552
Coefficient of Variation %	0.87%	2.52%	15.15%	5.63%

Table 18: Multi-Threaded Scores with CoV on Compute-Intensive Instance

#### 6.1.3.2 Price-per-Server Performance on Compute-Intensive Instance

Price-per-server performance value for each IaaS platform was determined by dividing the respective score by VM charge per hour. Table 19 shows the comparison of price-per-server performance values across all the IaaS cloud platforms. From the table it can be observed that the price-per-server performance value was higher for GCE. Therefore, GCE was the best-valued IaaS platform.

IaaS Cloud Platforms	Price-per-server performance value
Amazon EC2	13226.20
Google Compute Engine	20959.44
Microsoft Azure	9846.50
Rackspace	8053.86

Table 19: Price-per-Server Performance on Compute-Intensive Instance

## 6.2 File I/O Performance

The Dbench benchmark was used to test the file I/O performance. Two independent tests - standard file I/O test and file I/O performance overhead test, were performed on each IaaS platform across test sessions. The standard file I/O test was conducted by increasing the number of client processes (1, 2, 25, 50, 100, 200, and 300), a total of seven data points. File I/O performance overhead was tested by performing the file I/O operations with an increasing number of client processes (1, 2, 25, 50, 100, 200, and 300).

### 6.2.1 General Purpose Instances

Table 20 shows the comparison of standard file I/O throughput (MB/s) and file I/O performance overhead throughput (MB/s) across the IaaS cloud platforms. The throughput values listed in the table were obtained by calculating the median of individual throughput results captured across the sessions. Figure 7 shows the comparison of standard file I/O throughput (MB/s) across the IaaS cloud platforms on general purpose instance. Figure 8 shows the comparison of file I/O performance overhead throughput (MB/s) across the IaaS cloud platforms on general purpose instance.

IaaS Cloud Platforms	Standard file I/O throughput(MB/s)	File I/O overhead throughput(MB/s)
Amazon EC2	137.60	60.84
Google Compute Engine	181.64	69.64
Microsoft Azure	21.99	7.56
Rackspace	389.75	223.74

Table 20: File I/O (STD and OHD) Throughput on General Purpose Instance

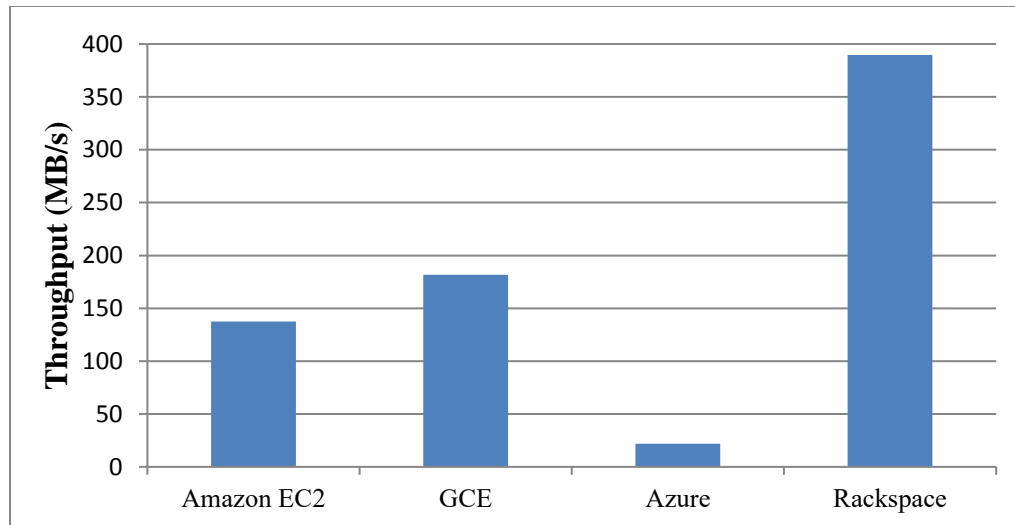


Figure 7: Standard File I/O Throughput on General Purpose Instance

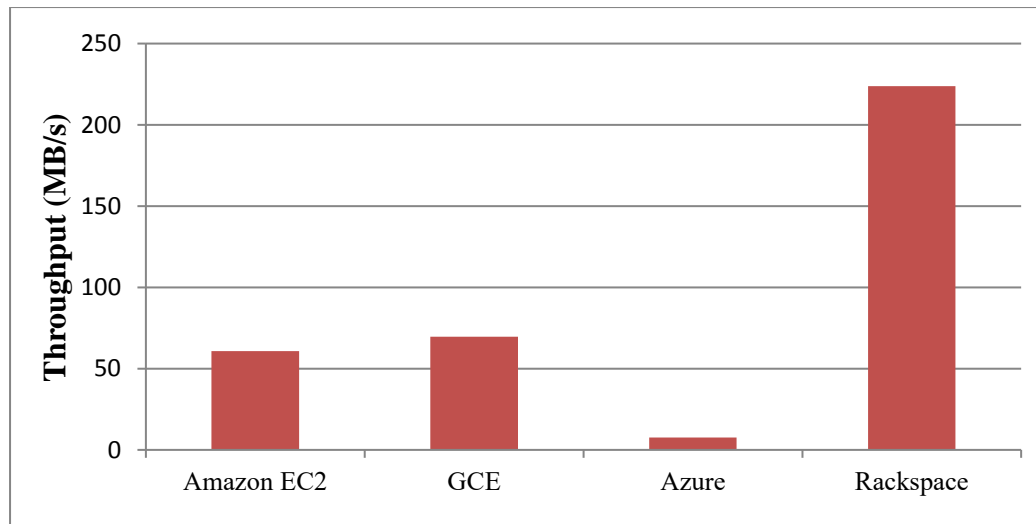


Figure 8: File I/O Overhead Throughput on General Purpose Instance

Rackspace delivered higher standard file I/O throughput because its general purpose instance is backed by SSD storage with RAID 10 support [Rackspace16]. Azure delivered the least performance because its general purpose instance is backed by local HDD storage [Posey15]. Figure 7 indicate that Rackspace's standard file I/O throughput is 17 times more than Azure's standard file I/O throughput. Figure 8 indicates that Rackspace's file I/O performance overhead throughput is 29.5 times more than Azure's file I/O performance overhead throughput.

#### 6.2.1.1 Performance Variability

For both standard file I/O and file I/O overhead tests, P-value was calculated for each pair of IaaS cloud platforms on the throughput results with different set of client parameters (50,100, 200, and 300) captured from four test sessions. Table 4 (See Appendix D for Statistical Analysis) shows the P-values for standard file I/O throughput and file I/O overhead throughput values on General purpose instance. The differences in the standard file I/O throughput values between five pair of IaaS cloud platforms - GCE Vs Rackspace, GCE Vs Azure, Amazon EC2 Vs Rackspace, Amazon EC2 Vs Azure, and Rackspace Vs Azure were found to be statistically significant with a P-value  $< 0.05$  across different client parameters -50,100, 200, and 300. The difference in the file I/O overhead throughput values between four pair of IaaS cloud platforms - GCE Vs Rackspace, GCE Vs Azure, Amazon EC2 Vs Rackspace, and Rackspace Vs Azure were found to be statistically significant across different client parameters - 50, 100, 200, and 300.

Table 21 shows the standard file I/O throughput results of each IaaS platform throughout the test period with CoV displaying the variability in performance. Table 22 shows the file I/O overhead throughput of each IaaS platform throughout the test period with CoV indicating the variability in performance. In both the tests, GCE was the most consistent IaaS platform with the least CoV, possibly due to its persistent disk storage that offers sustained performance [GCE17B]. Azure displayed the high variability in performance, possibly due to the load conditions and noisy neighbors residing on the same physical host [Makroo16].

Sessions	Amazon EC2 (MB/s)	Google Compute Engine (MB/s)	Microsoft Azure (MB/s)	Rackspace (MB/s)
Session-1	168.93	188.12	36.28	393.26
Session-2	103.49	185.25	10.37	386.25
Session-3	182.19	178.20	18.06	384.42
Session-4	129.80	175.32	28.38	435.08
Coefficient of Variation %	24.70%	3.28%	48.92%	5.97%

Table 21: Standard File I/O Throughput with CoV on General Purpose Instance

Sessions	Amazon EC2 (MB/s)	Google Compute Engine (MB/s)	Microsoft Azure (MB/s)	Rackspace (MB/s)
Session-1	60.92	70.06	20.82	222.01
Session-2	61.21	70.50	4.94	224.29
Session-3	61.05	69.59	8.40	218.96
Session-4	55.41	66.13	14.48	255.61
Coefficient of Variation %	4.74%	2.89%	57.50%	7.41%

Table 22: File I/O Overhead Throughput with CoV on General Purpose Instance

#### 6.2.1.2 Price-per-Performance on General Purpose Instance

Price-per-file I/O performance value for each IaaS platform was determined by dividing the respective throughput result by VM charge per hour. Table 23 shows the comparison of price-per-file I/O performance values across all the IaaS cloud platforms. The price-per-file I/O performance value was higher for GCE. Therefore, GCE was the best-valued IaaS platform.

IaaS Cloud Platforms	Price-per-file I/O performance value
Amazon EC2	833.42
Google Compute Engine	2401.38
Microsoft Azure	79.58
Rackspace	1912.31

Table 23: Price-per-File I/O Performance on General Purpose Instance

## 6.2.2 Memory-Intensive Instances

Table 24 shows standard file I/O throughput (MB/s) and file I/O performance overhead throughput (MB/s) across the IaaS cloud platforms. The throughput values listed in the table were obtained by calculating the median of individual throughput results captured. Figure 9 shows the comparison of standard file I/O throughput (MB/s) across the IaaS cloud platforms on memory-intensive instance. Figure 10 shows the comparison of file I/O performance overhead throughput (MB/s) across the IaaS cloud platforms on memory-intensive instance.

IaaS Cloud Platforms	Standard file I/O throughput(MB/s)	File I/O overhead throughput(MB/s)
Amazon EC2	305.28	114.55
Google Compute Engine	247.85	75.42
Microsoft Azure	41.69	17.99
Rackspace	1154.39	206.49

Table 24: File I/O (STD and OHD) Throughput on Memory-Intensive Instance

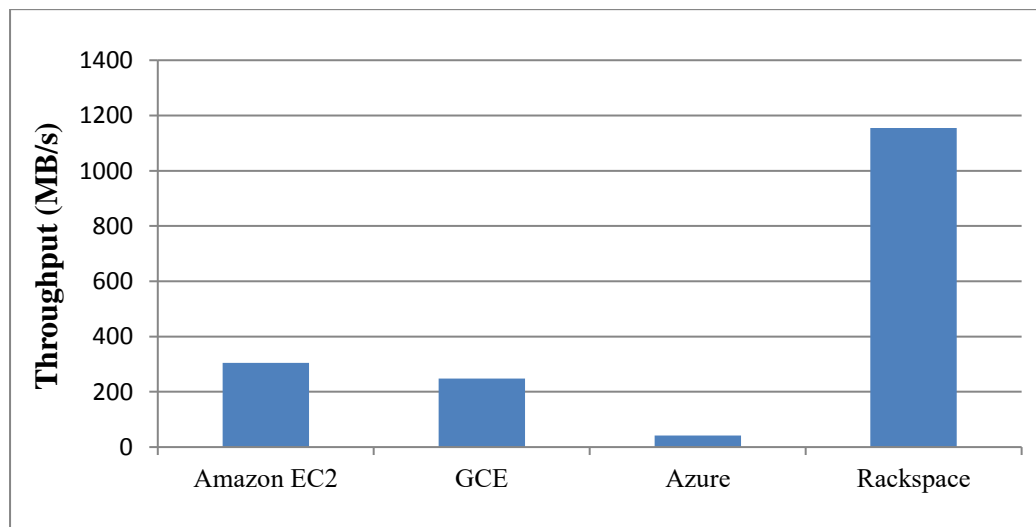


Figure 9: Standard File I/O Throughput on Memory-Intensive Instance



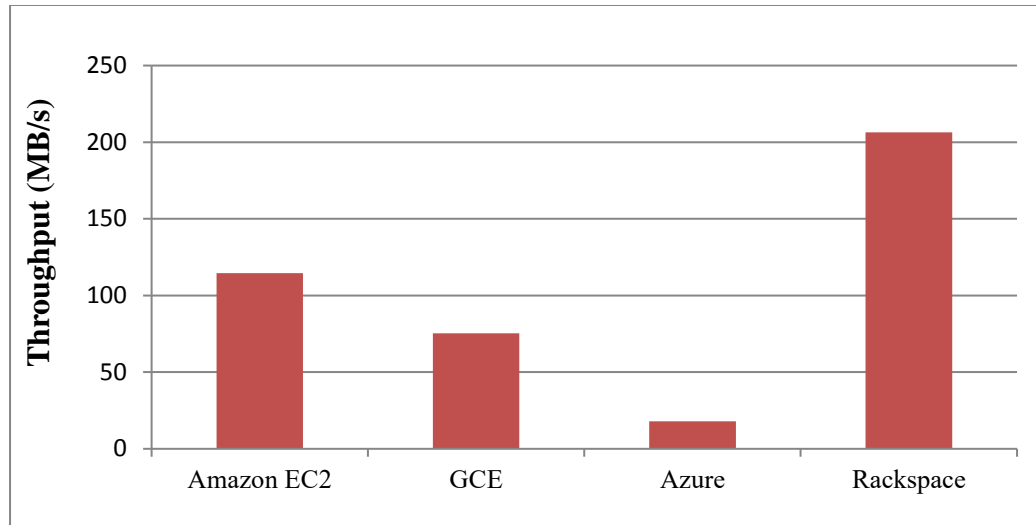


Figure 10: File I/O Overhead Throughput on Memory-Intensive Instance

Rackspace delivered significantly higher throughput because all the disk storage for this instance is on Cloud Block Storage (CBS). The Rackspace Cloud Block Storage provides persistent block-level storage volumes that offer reliable and high performance SSD [Posey15]. Figure 9 indicate that Rackspace standard file I/O throughput is 27.6 times more than the Azure's standard file I/O throughput. Figure 10 indicates that Rackspace file I/O performance overhead throughput is 11.4 times more than that of Azure.

#### 6.2.2.1 Performance Variability

For both standard file I/O and file I/O overhead tests, P-value was calculated for each pair of IaaS cloud platforms on the throughput results with different set of client parameters (50,100, 200, and 300) captured all the four test sessions. Table 5 (See Appendix D for Statistical Analysis) shows the P-values for standard file I/O throughput and file I/O overhead throughput values on Memory-intensive instance. The difference in the

standard file I/O throughput and file I/O overhead throughput values between all pairs of IaaS cloud platforms were found to be statistically significant with a P-value  $< 0.05$  across different client parameters -50,100, 200, and 300.

Table 25 shows the standard file I/O throughput results of each IaaS platform throughout the test period with CoV displaying the variability in performance. Table 26 shows the file I/O overhead throughput of each IaaS platform throughout the test period with CoV indicating the variability in performance. In both the tests, Amazon EC2 was the most consistent provider with the least CoV. Azure displayed the highest performance variability, possibly due to the load conditions and noisy neighbors residing on the same physical host [Makroo16].

Sessions	Amazon EC2 (MB/s)	Google Compute Engine (MB/s)	Microsoft Azure (MB/s)	Rackspace (MB/s)
Session-1	301.49	263.02	52.76	1104.61
Session-2	306.99	245.16	34.98	1183.22
Session-3	303.57	242.28	36.16	1156.00
Session-4	316.91	250.54	38.74	1152.77
Coefficient of Variation %	2.22%	3.67%	20.21%	2.84%

Table 25: Standard File I/O Throughput with CoV on Memory-Intensive Instance

Sessions	Amazon EC2 (MB/s)	Google Compute Engine (MB/s)	Microsoft Azure (MB/s)	Rackspace (MB/s)
Session-1	114.53	75.44	18.36	198.47
Session-2	114.59	75.29	20.40	206.59
Session-3	114.45	75.81	17.08	210.16
Session-4	114.56	75.41	17.61	206.40
Coefficient of Variation %	0.05%	0.30%	7.93%	2.40%

Table 26: File I/O Overhead Throughput with CoV on Memory-Intensive Instance

#### 6.2.2.2 Price-per-File I/O Performance on Memory-Intensive Instance

Price-per-file I/O performance for each provider was determined by dividing the respective throughput by VM charge per hour. Table 27 shows the comparison of price-per-file I/O performance values across all the IaaS cloud platforms. The price-per-file I/O performance value was higher for GCE. Therefore, GCE was the best-valued IaaS platform.

IaaS Cloud Platforms	Price-per-file I/O performance value
Amazon EC2	254.56
Google Compute Engine	426.10
Microsoft Azure	41.74
Rackspace	375.98

Table 27: Price-per-File I/O Performance on Memory-Intensive Instance

### 6.2.3 Compute-Intensive Instances

Table 28 shows the comparison of standard file I/O throughput (MB/s) and file I/O performance overhead throughput (MB/s) across the four IaaS cloud platforms. The throughput values listed in the table were obtained by calculating the median of individual throughput results captured across the sessions. Figure 11 shows the comparison of standard file I/O throughput (MB/s) across the four IaaS cloud platforms on compute-intensive instance. Figure 12 shows the comparison of file I/O performance overhead throughput (MB/s) across the four IaaS cloud platforms on compute-intensive instance.

IaaS Cloud Platforms	Standard file I/O throughput(MB/s)	File I/O overhead throughput(MB/s)
Amazon EC2	293.94	79.14
Google Compute Engine	216.61	75.21
Microsoft Azure	33.56	17.71
Rackspace	1332.47	210.81

Table 28: File I/O (STD and OHD) Throughput on Compute-Intensive Instance

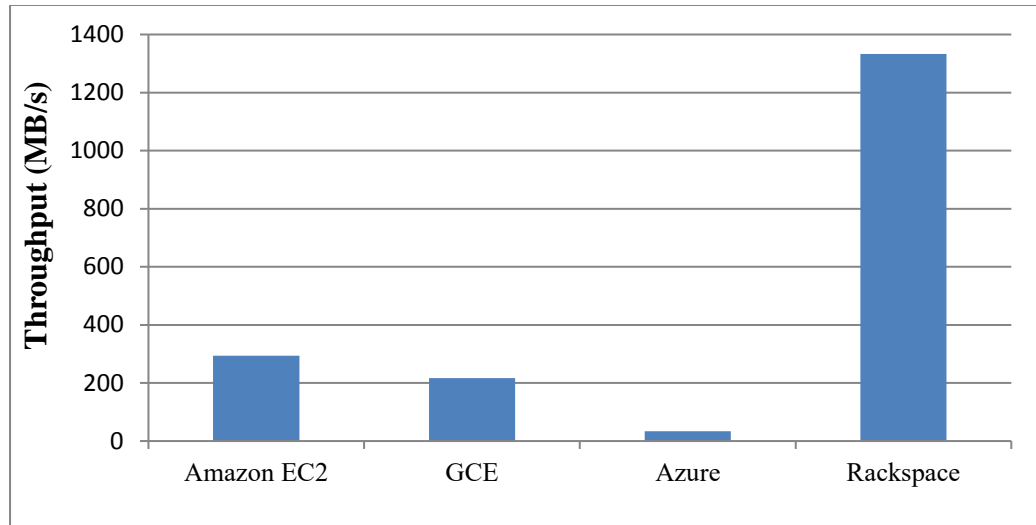


Figure 11: Standard File I/O Throughput on Compute-Intensive Instance

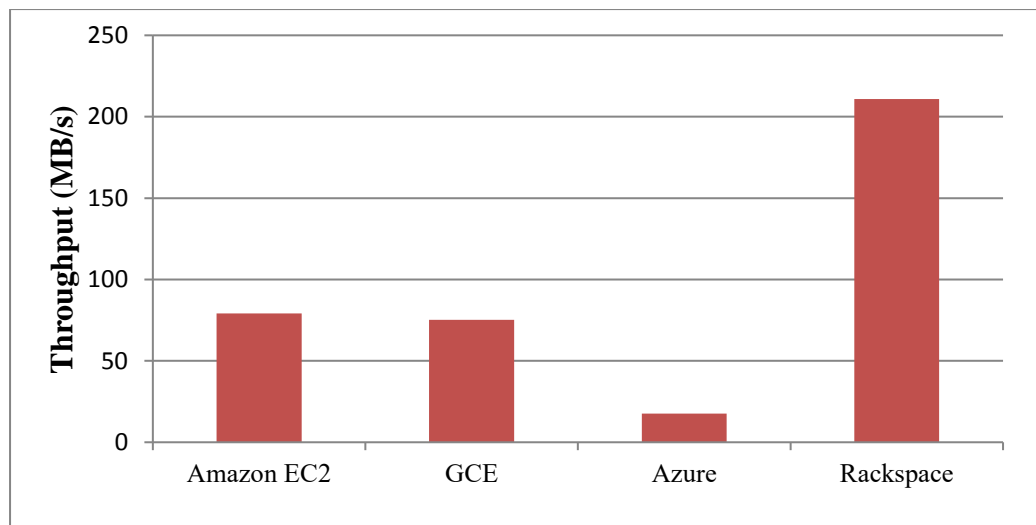


Figure 12: File I/O Overhead Throughput on Compute-Intensive Instance

Rackspace delivered significantly higher throughput because all disk storage is on Cloud Block Storage. The Rackspace Cloud Block Storage provides persistent block-level storage volumes that offer reliable high performance SSD [Posey15]. Figure 11 indicate that Rackspace's standard file I/O throughput is 39.6 times more than Azure's standard

file I/O throughput. Figure 12 indicate that Rackspace's file I/O performance overhead throughput is 11.9 times more than that of Azure.

#### 6.2.3.1 Performance Variability

For both standard file I/O and file I/O overhead tests, P-values were calculated for each pair of IaaS cloud platforms on the throughput results with different set of client parameters (50,100, 200, and 300) captured from four test sessions. Table 6 (See Appendix D for Statistical Analysis) shows the P-values for standard file I/O throughput and file I/O overhead throughput values on Compute-intensive instance. For both standard file I/O and file I/O overhead throughput tests, the differences in the throughput values between four pairs of IaaS cloud platforms - GCE Vs Rackspace, GCE Vs Azure, Amazon EC2 Vs Rackspace, and Rackspace Vs Azure were statistically significant with a P-value  $< 0.05$  across different client parameters -50,100, 200, and 300.

Table 29 shows the standard file I/O throughput results of each IaaS platform throughout the test period with CoV indicating the variability in performance. Table 30 shows the file I/O overhead throughput of each IaaS platform throughout the test period with CoV indicating the variability in performance. In both the tests, GCE was the most consistent IaaS platform with least CoV. Amazon EC2 displayed high variability in the performance, possibly due to the load conditions and noisy neighbors residing on the same physical host [Makroo16].

Sessions	Amazon EC2 (MB/s)	Google Compute Engine (MB/s)	Microsoft Azure (MB/s)	Rackspace (MB/s)
Session-1	349.44	220.39	34.52	1325.91
Session-2	351.16	212.83	28.22	1328.37
Session-3	221.84	221.44	25.06	1502.46
Session-4	296.66	212.52	38.56	1336.57
Coefficient of Variation %	19.95%	2.2%	19.26%	6.28%

Table 29: Standard File I/O Throughput with CoV on Compute-Intensive Instance

Sessions	Amazon EC2 (MB/s)	Google Compute Engine (MB/s)	Microsoft Azure (MB/s)	Rackspace (MB/s)
Session-1	349.44	220.39	34.52	1325.91
Session-2	351.16	212.83	28.22	1328.37
Session-3	221.84	221.44	25.06	1502.46
Session-4	296.66	212.52	38.56	1336.57
Coefficient of Variation %	19.95%	2.2%	19.26%	6.28%

Table 30: File I/O Overhead Throughput with CoV on Compute-Intensive Instance

#### 6.2.3.2 Price-per-File I/O Performance on Compute-Intensive Instance

Price-per-file I/O performance for each IaaS platform was determined by dividing the respective throughput by VM charge per hour. Table 31 shows the comparison of price-per-file I/O performance values across all IaaS cloud platforms. The price-per-file I/O performance value was higher for Rackspace. Therefore, Rackspace was the best-valued IaaS platform.

IaaS Cloud Platforms	Price-per-file I/O performance value
Amazon EC2	179.864
Google Compute Engine	302.048
Microsoft Azure	27.543
Rackspace	383.849

Table 31: Price-per-File I/O Performance on Compute-Intensive Instance

### 6.3 Network Performance

The Iperf benchmarking tool was used to test the network performance between two virtual machines. Load testing was performed by increasing the number of parallel connections (1, 50, and 128) from the client VM to server VM. The individual throughput results were captured and the median was calculated.

#### 6.3.1 General Purpose Instances

Table 32 and Figure 13 show the comparison of network throughput (Mbps) across the IaaS cloud platforms on general purpose instance. The throughput values listed in the table were obtained by calculating the median of individual throughput results captured across the test sessions.

IaaS Cloud Platforms	Network Throughput (Mbps)
Amazon EC2	87.55
Google Compute Engine	1986.57
Microsoft Azure	487.82
Rackspace	101.25

Table 32: Network Throughput on General Purpose Instance



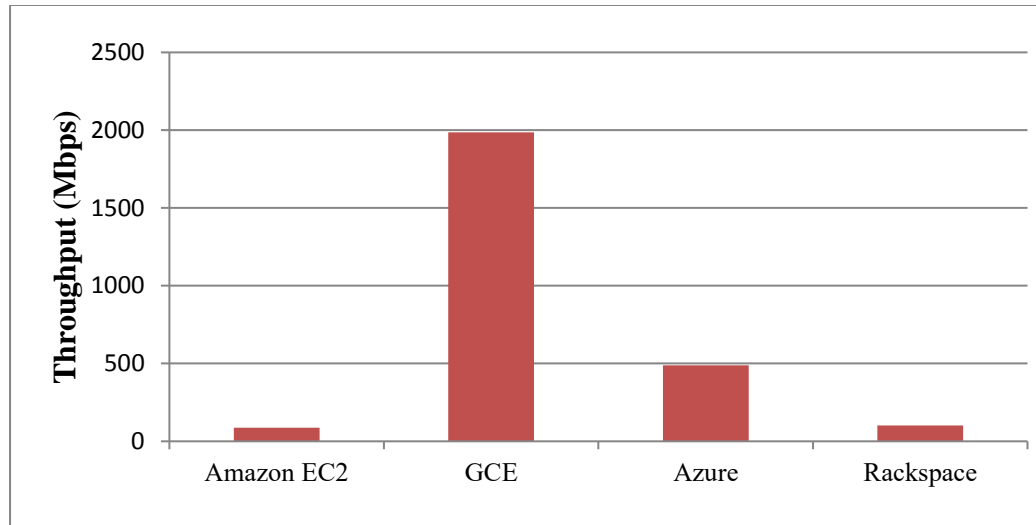


Figure 13: Network Throughput on General Purpose Instance

Figure 13 indicates that GCE delivered the highest network throughput compared to other IaaS cloud platforms. This is because GCE allocates 2Gbps bandwidth per CPU core for peak performance [GCE17C]. The throughput delivered by GCE was 22 times that of Amazon EC2 and 19.6 times that of Rackspace. Amazon EC2 and Rackspace delivered the least throughput because of their inherent bandwidth limitations [Mytton14]. However, all the IaaS cloud platforms delivered throughput close to their respective advertised bandwidth.

#### 6.3.1.1 Performance Variability

Table 7 (See Appendix D for Statistical Analysis) shows P-values calculated for each pair of IaaS cloud platforms on the throughput results with different set of client parameters (50 and 128) captured from four test sessions. The difference in throughput values

between Amazon EC2 Vs Rackspace was found to be statistically insignificant with a P-value  $> 0.05$  across different sets of client parameters (50 and 128).

Table 33 shows the network throughput results of each provider throughout the test period with CoV showing the variability in performance. Rackspace was the most consistent provider with the least CoV. Amazon EC2 displayed the highest variability in performance, possibly due to the burstable CPU core [Azavea14].

Sessions	Amazon EC2 (Mbps)	Google Compute Engine (Mbps)	Microsoft Azure (Mbps)	Rackspace (Mbps)
Session-1	113.42	1985.63	489.33	101.14
Session-2	76.91	1988.21	488.78	101.27
Session-3	98.19	1984.3	454.46	101.23
Session-4	72.37	1987.44	477.63	101.3
Coefficient of Variation %	33.59%	3.54%	0.46%	0.25%

Table 33: Network Throughput with CoV on General Purpose Instance

#### 6.3.1.2 Price-per-Network Performance on General Purpose Instance

Price-per-network performance for each provider was determined by dividing the respective throughput by VM charge per hour. Table 34 shows the comparison of price-per-network performance values across all the IaaS cloud platforms. The price-per-network performance value was higher for GCE. Therefore, GCE was the best-valued IaaS platform.

IaaS Cloud Platforms	Price-per-network performance value
Amazon EC2	1199.32
Google Compute Engine	68502.41
Microsoft Azure	5134.95
Rackspace	865.38

Table 34: Price-per-Network Performance on General Purpose Instance

### 6.3.2 Memory-Intensive Instances

Table 35 and Figure 14 show the comparison of network throughput (Mbps) across the IaaS cloud platforms on memory-intensive instance. The throughput values listed in the table were obtained by calculating the median of individual throughput results captured across the sessions.

IaaS Cloud Platforms	Network Throughput(Mbps)
Amazon EC2	695.06
Google Compute Engine	7855.56
Microsoft Azure	2905.34
Rackspace	6470.23

Table 35: Network Throughput on Memory-Intensive Instance

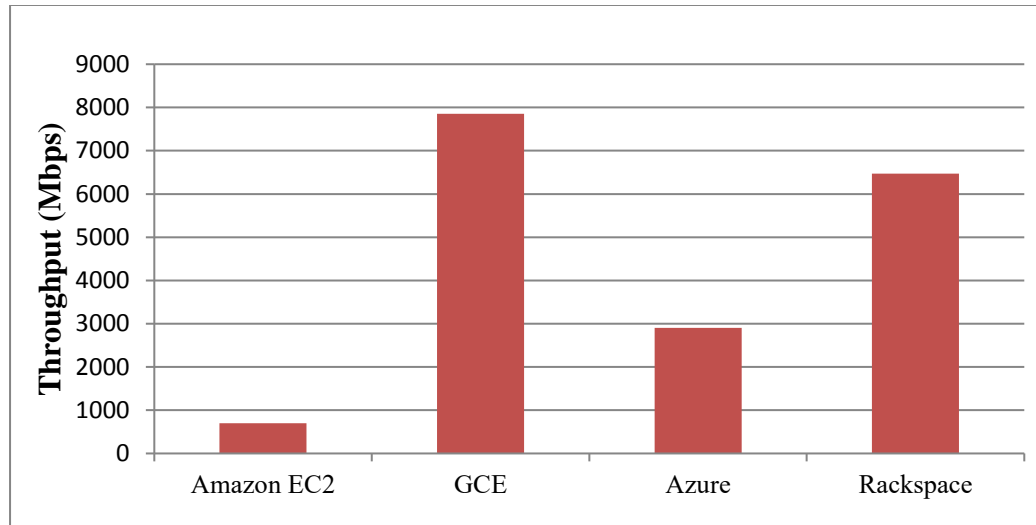


Figure 14: Network Throughput on Memory-Intensive Instance

Figure 14 shows that GCE delivered the highest network throughput compared to other IaaS cloud platforms on memory-intensive instance. This is because GCE allocates 8Gbps bandwidth per CPU core for peak performance and it achieved the advertised throughput [GCE17C]. The throughput delivered by GCE was 11 times that of Amazon EC2. Amazon EC2 delivered the least throughput because the network bandwidth in Amazon's memory-intensive instance is throttled at 500 Mbps [AmazonEC217B].

#### 6.3.2.1 Performance Variability

Table 8 (See Appendix D for Statistical Analysis) shows P-values calculated for each pair of IaaS cloud platforms on the throughput results with different set of client parameters (50 and 128) captured from four test sessions. The differences in throughput values between each pair of IaaS cloud platforms were found to be statistically significant with a P-value < 0.05 across different client parameters.

Table 36 shows the network throughput results of each IaaS platform throughout the test period with CoV indicating the variability in performance. Amazon EC2 was the most consistent IaaS platform with the least CoV, and Rackspace was the most variable IaaS platform.

Sessions	Amazon EC2 (Mbps)	Google Compute Engine (Mbps)	Microsoft Azure (Mbps)	Rackspace (Mbps)
Session-1	695.11	7825.5	2894.87	6131.73
Session-2	695.02	7847.95	2736.85	6447.53
Session-3	695.07	7729.41	2935.05	6492.93
Session-4	695.04	7920.75	2953.13	6826.49
Coefficient of Variation %	0.08%	4.58%	1.37%	5.55%

Table 36: Network Throughput with CoV on Memory-Intensive Instance

#### 6.3.2.2 Price-per-Network Performance on Memory-Intensive Instance

Price-per-network performance for each provider was determined by dividing the respective throughput by VM charge per hour. Table 37 shows the comparison of price-per-network performance values across all the IaaS cloud platforms. The price-per-network performance value was higher for GCE. Therefore, GCE was the best-valued IaaS platform.

Cloud providers	Price-per-network performance value
Amazon EC2	1544.58
Google Compute Engine	44381.69
Microsoft Azure	6740.93
Rackspace	11781.19

Table 37: Price-per-Network Performance on Memory-Intensive Instance

### 6.3.3 Compute-Intensive Instances

Table 38 and Figure 15 show the comparison of network throughput (Mbps) across the IaaS cloud platforms. The throughput values listed in the table are obtained by calculating the median of individual throughput results captured across the sessions.

Cloud providers	Network Throughput(Mbps)
Amazon EC2	2481.39
Google Compute Engine	14401.56
Microsoft Azure	4378.07
Rackspace	573.725

Table 38: Network Throughput on Compute-Intensive Instance

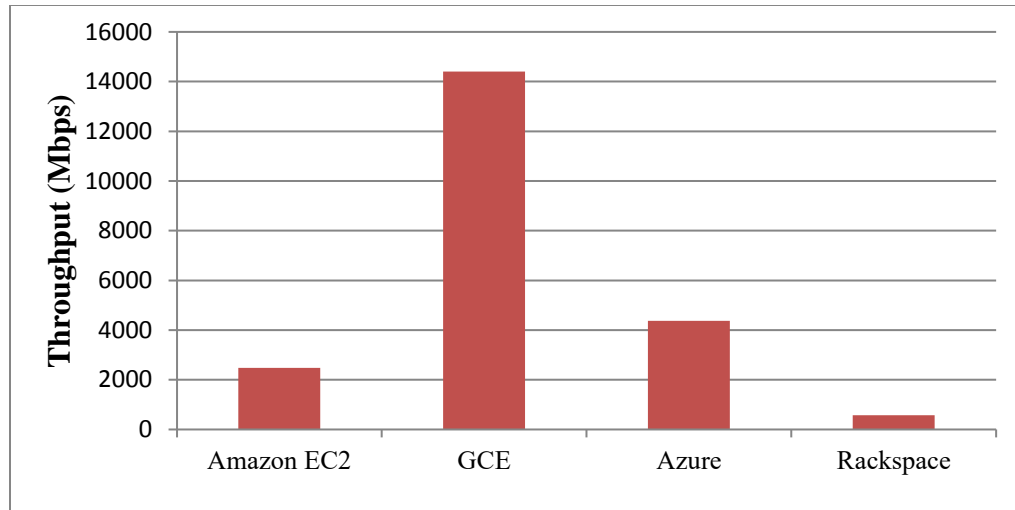


Figure 15: Network Throughput on Compute-Intensive Instance

Figure 15 indicates that GCE delivered the highest network throughput compared to other IaaS cloud platforms on compute-intensive instance. This is because GCE allocates 16Gbps bandwidth per CPU core for peak performance [GCE17C]. The throughput delivered by GCE was 25 times that of Rackspace. Rackspace delivered the least network performance and the result was much lower than its advertised network bandwidth (1.3Gbps) [Rackspace16].

#### 6.3.3.1 Performance Variability

Table 9 (See Appendix D for Statistical Analysis) shows P-values calculated for each pair of IaaS cloud platforms on the throughput results with different set of client parameters (50 and 128) captured from four test sessions. The differences in throughput values between each pair of IaaS cloud platforms were found to be statistically significant with a P-value  $< 0.05$  across different client parameters.

Table 39 shows the network throughput results of each IaaS platform throughout the test period with CoV indicating the variability in performance. The Coefficient of Variation was significantly low for Amazon EC2. Therefore, Amazon EC2 was the most consistent IaaS platform. Microsoft Azure demonstrated the highest variability in performance when compared to other IaaS cloud platforms.

Sessions	Amazon EC2 (Mbps)	Google Compute Engine (Mbps)	Microsoft Azure (Mbps)	Rackspace (Mbps)
Session-1	2481.39	15205.61	5914.13	573.93
Session-2	2481.39	15254.03	4110.3	571.95
Session-3	2481.39	14370.43	3866.47	621.6
Session-4	2481.37	13720.48	5349.06	556.74
Coefficient of Variation %	0.01%	25.66%	26.43%	16.11%

Table 39: Network Throughput with CoV on Compute-Intensive Instance

#### 6.3.3.2 Price-per-Network Performance on Compute-Intensive Instance

Price-per-network performance for each IaaS platform was determined by dividing the respective throughput by VM charge per hour. Table 40 shows the comparison of price-per-network performance values across all the IaaS cloud platforms. The price-per-network performance value was higher for GCE. Therefore, GCE was the best-valued IaaS platform.



IaaS Cloud Platforms	Price-per-network performance value
Amazon EC2	5639.52
Google Compute Engine	57837.59
Microsoft Azure	6808.82
Rackspace	1044.66

Table 40: Price-per-Network Performance on Compute-Intensive Instance

## Chapter 7

### CONCLUSIONS AND FUTURE WORK

The conclusions concerning the best IaaS platform for different application workloads were drawn from three criteria: performance, performance variability, and the price-per-performance. The detailed analysis in Chapter 6 illustrates the differences among platforms from both the performance and pricing perspectives. Performance and performance variability of IaaS cloud platforms with respect to server, file I/O, and network utilization were thoroughly examined using system-level benchmarks: Unixbench, Dbench, and Iperf, respectively. This analysis can help organizations select an appropriate platform that fits their application performance requirements. The combination of performance assessment and pricing was used to identify the most economical IaaS platform delivering the best performance.

Server performance testing showed that there were substantial differences in the performance of different IaaS cloud platforms. Benchmark results revealed that server performance depends on many factors such as CPU type and clock speed. Previous research [Lenk11] also concluded that there was strong relation between the server performance and underlying CPU architecture. The factors that are out of the user's control, such as noisy neighbors, hardware heterogeneity, and virtualization overhead, impacted the performance and accounted for the performance fluctuations. Previous research [Leitner16] also concluded that performance depends on factors such as

underlying hardware architecture and multi-tenancy. File I/O performance tests indicated that Rackspace's faster SSD and Cloud Block Storage (CBS) delivered the highest performance on all the tested VMs. In addition to the observed performance advantage, SSD storage delivered consistency by reducing I/O contention. Network performance benchmarking revealed that Google Compute Engine offered the highest overall performance results on all the tested instances due to its global networking architecture. The lowest observed network performance of Amazon EC2 was due to the bandwidth throttling of the IaaS cloud platforms. The price-per-performance analysis showed that Google Compute Engine displayed best price-per-performance value on all the benchmark tests. The independent customizable server offerings of Google Compute Engine let the users customize an optimal VM configuration for their application workload, thereby resulting in maximum cost savings. Researchers also have concluded that GCE is the least expensive platform for most of the computing requirements [Kihal12].

## 7.1 Further Research

In this thesis, the performance capabilities of four IaaS cloud platforms were evaluated at the system-level to identify the best cloud provider. However, there is a significant scope in extending the research to address several others factors that can contribute to the cloud platform deployment decision.

The primary components of virtual servers are virtual cores, memory, storage, and network. The scope of the study was limited to benchmarking server, file I/O, and

network performance. Hence, the scope of research could be extended to include an assessment of the memory performance of IaaS cloud platforms.

This study was limited to performing benchmarking on equivalent general purpose instances, memory-intensive instances, and compute-intensive instances only. The study can serve as a reference point for further studies. One such study might extend the scope by benchmarking other instance types, such as I/O-intensive and network-intensive instances.

In this study, benchmarking was only focused on the throughput. Extension to this study could include consideration of other metrics that can impact application performance.

One such example is latency, which is another important measure to gauge the performance of memory, storage and network. Therefore, the scope of benchmarking could be extended to measure the latency along with the throughput.

## REFERENCES

### Print Publications:

[CloudSpectator15]

CloudSpectator, "The Successful Executive's Guide To Selecting The Right Cloud Provider", Technical Report, 2015.

[Jackson11]

Jackson, K.R., L. Ramakrishnan, K. Muriki, S. Cannon, S. Cholia, J. Shalf, H.J. Wasserman, and N.J. Wright., "Performance Analysis of High Performance Computing Applications on the Amazon Web Services cloud", Proceedings of the IEEE Second International Conference on Cloud Computing Technology and Science (Cloudcom), Indiana, pages 159-168, February 2011.

[Kihal12]

Kihal, S.E., C. Schlereth, and B. Skiera., "Price Comparison for Infrastructure as a Service", European Conference on Information Systems, Barcelona, Spain, 2012.

[Leitner16]

Leitner, P and J. Cito, "Patterns in the Chaos—A Study of Performance Variation and Predictability in Public IaaS Clouds", ACM Transactions on Internet Technology, vol. 16, pages 1-23, ISSN 15335399, April 2016.

[Lenk11]

Lenk, A., M. Menzel, J. Lipsky, S. Tai, and P. Offermann., "What are you paying for? performance benchmarking for Infrastructure-as-a-Service offerings", Proceedings of IEEE Fourth International Conference on Cloud Computing (CLOUD), Washington, D.C., pages 484-491, July 2011.

[Mell11]

Mell, P and T. Grance, "The NIST Definition of Cloud Computing", Recommendations of National Institute of Standards and Technology, US. Department of Commerce, Special Publication, September 2011.

[Makroo16]

Makroo, A and D. Dahiya, "A systematic approach to deal with noisy neighbor in cloud infrastructure", Indian Journal of Science and Technology, vol.9 (19), pages 4-7, ISSN 0974-5645, May 2016.

[Ortiz11]

Ortiz, S.Jr. "The Problem with Cloud-Computing Standardization ", IEEE Computer Society, vol.44, pages 13-16, July 2011.

[Persico15]

Persico, V., P. Marchetta, A.Botta, and A. Pescapé, " Measuring Network Throughput in the Cloud: The case of AmazonEC2", Elsevier, Computer Networks, Special Issue on Cloud Networking and Communications II, vol.93, pages 408-422, December 24, 2015.

[Schad10]

Schad, J., J. Dittrich, and J. QuijaneRuiz., "Runtime measurements in the cloud: observing, analyzing, and reducing variance", Proceedings of the VLDB Endowment, vol.3, pages 460-471, September 2010.

[Shawish14]

Shawish, A and M. Salama., "Cloud Computing: Paradigms and Technologies", Inter-cooperative Collective Intelligence: Techniques and Applications, Studies in Computational Intelligence 495, Springer, pages 39-67, 2014.

#### Electronic Sources:

[AmazonEC217A]

services, amazon web, "EC2 Instance Types", <https://aws.amazon.com/ec2/instance-types/>, last accessed May 18, 2017.

[AmazonEC217B]

services, amazon web, "Amazon EC2 Instance Configuration", <https://aws.amazon.com/ec2/instance-types/>, last accessed May 18, 2017.

[Azavea14]

Azavea and SciStarter, "A Distribution Data Collection Platform for Citizen Science", Citizen Science Data Factory report, February 2014.

[Barr14]

Barr, J., "New Compute-Optimized EC2 Instances", <https://aws.amazon.com/blogs/aws/new-c4-instances/>, last revision November 13, 2014, last accessed May 23, 2017.

[Combs12]

Combs, K., "What is Infrastructure as a Service (IaaS)?", <https://blogs.technet.microsoft.com/privatecloud/2012/04/04/what-is-infrastructure-as-a-service-iaas/>, last revision April 2012, last accessed May 18, 2017.

[CloudHarmony14]

CloudHarmony, "CenturyLink Hyperscale Testing",  
[https://www.ctl.io/lp/resources/Cloud-Harmony\\_CenturyLink-Hyperscale\\_Report.pdf](https://www.ctl.io/lp/resources/Cloud-Harmony_CenturyLink-Hyperscale_Report.pdf), last revision April 15, 2014, last accessed May 22, 2017.

[Cook08]

Cook, R., "The top 10 causes of slow data backups", SearchDataBackup,  
<http://searchdatabackup.techtarget.com/tip/The-top-10-causes-of-slow-data-backups>, last revision July 2008, last accessed May 24, 2017.

[Dbench17]

Dbench, "Welcome to the DBENCH web pages", <https://dbench.samba.org/>, last accessed May 22, 2017.

[Github17]

Github, "byte-Unixbench", <https://github.com/meteorfox/byte-unixbench>, last accessed May 22, 2017.

[GCE17A]

Google Cloud Platform, "Machine types- Compute Engine Documentation",  
<https://cloud.google.com/compute/docs/machine-types>, last accessed May 18, 2017.

[GCE17B]

Google Cloud Platform, "Storage Options- Compute Engine Documentation",  
<https://cloud.google.com/compute/docs/disks/>, last accessed May 18, 2017.

[GCE17C]

Google Cloud Platform, "VPC Networking and Firewalls - Compute Engine Documentation ", <https://cloud.google.com/compute/docs/networks-and-firewalls>, last accessed May 18, 2017.

[Iperf17]

Iperf, "Iperf- The ultimate speed test tool for TCP, UDP and SCTP", <https://iperf.fr/iperf-doc.php>, last accessed May 22, 2017.

[Kajeepeta10]

Kajeepeta, S., "Multi-tenancy in the cloud: Why it matters", Computerworld,  
<http://www.computerworld.com/article/2517005/data-center/multi-tenancy-in-the-cloud--why-it-matters.html>, last revision April 2012, last accessed October 21, 2016.

[Keating16]

Keating, C., "How to Test Storage Performance in Cloud Environments",  
<https://www.zadarastorage.com/blog/tech-corner/how-to-test-storage-performance-in-cloud-environments/>, last revision August 2016, last accessed May 25, 2017.

[Kwa14]

Kwa, K., "Basic Tier Virtual Machines", <https://azure.microsoft.com/en-us/blog/basic-tier-virtual-machines-2/>, last revision April 28, 2014, last accessed May 25, 2017.

[Lavnevich16]

Lavnevich, N., "Project Success Plan: Cloud Migration", SearchCloudComputing, TechTarget, August 24, 2016.

[Liu16]

Liu, A., "Geographic Cloud Performance Comparison", <https://cloudspectator.com/geographic-cloud-performance-comparison/>, last revision July 20, 2016, last accessed May 23, 2017.

[Mytton14]

Mytton, D., "Need For speed: Testing the networking performance of The top 4 cloud providers", <https://gigaom.com/2014/04/12/need-for-speed-testing-the-networking-performance-of-the-top-4-cloud-providers/>, last revision April 12, 2014, last accessed May 24, 2017.

[Nottingham16]

Nottingham, C., "Sizes for Virtual machines in Azure", <https://docs.microsoft.com/en-us/azure/virtual-machines/virtual-machines-windows-sizes>, last accessed November 8, 2016.

[Posey15]

Posey, B., "2016 Public Cloud Provider Buyer's Guide", TechTarget, May 2015, last accessed May 22, 2017.

[Profitbricks13]

Profitbricks, "Secret World of Cloud IaaS Pricing: How to Compare Apples and Oranges among Cloud Providers", [https://www.profitbricks.co.uk/sites/default/files/PB\\_Secret-World-Of-Cloud-Computing-Pricing-How-To-Compare-Providers.pdf](https://www.profitbricks.co.uk/sites/default/files/PB_Secret-World-Of-Cloud-Computing-Pricing-How-To-Compare-Providers.pdf), last revision August 2013, last accessed May 23, 2017.

[Pwc11]

Pwc, "Protecting your brand in the cloud: Transparency and trust through enhanced reporting", <http://www.pwc.com/us/en/third-party-assurance/assets/protecting-your-brand-in-the-cloud-enhancing-transparency.pdf>, November 2011.

[Perry10]

Perry, G., "Evaluating cloud computing services: Criteria to consider", <http://searchcloudcomputing.techtarget.com/feature/Evaluating-cloud-computing-services-Criteria-to-consider>, last revision November 2010, last accessed May 25, 2017.



[Rossa14]

Rossa, V.D., "How will the IT organization change because of cloud innovation?", Cloud computing news, IBM, <https://www.ibm.com/blogs/cloud-computing/2014/10/will-organization-change-cloud-innovation/>, last revision October 18, 2014, last accessed May 24, 2017.

[Rackspace16]

Rackspace, "Virtual Cloud Servers Powered By OpenStack", <https://www.rackspace.com/en-us/cloud/servers>, last accessed November 8, 2016.

[Tata11]

Tata Communications, "Infrastructure-as-a-Service: Fulfilling the Promise of Cloud Computing", [http://www.tatacommunications.com/downloads/whitepapers/IaaS\\_white\\_paper.pdf](http://www.tatacommunications.com/downloads/whitepapers/IaaS_white_paper.pdf), 2011.

[Wise11]

Wise, C., "Achieving High Performance: The Value of Benchmarking", <http://corporateplanningresources.com/pdfs/Accenture-Achieving-High-Performance-The-Value-of-Benchmarking.pdf>, 2011.

[Wells15]

Wells, E., "Cloud: The Essential Buyer's Guide", [http://cloud.report/Resources/Whitepapers/0d2d515b-b8b6-4c96-9dd0-d387773aa112\\_THE%20ESSENTIAL%20BUYERS%20GUIDE.pdf](http://cloud.report/Resources/Whitepapers/0d2d515b-b8b6-4c96-9dd0-d387773aa112_THE%20ESSENTIAL%20BUYERS%20GUIDE.pdf), November 2015.

## APPENDIX A

### Unixbench Installation and Execution

Launch the instance of same category (general purpose instance, memory-intensive instance, and compute-intensive instance) on Amazon EC2, Google Compute Engine, Microsoft Azure, and Rackspace. Use PuTTY or any other SSH client to connect to the public DNS of the instance.

The process for configuring and executing Unixbench is described in following steps:

1. Before installing the benchmark, compiler packages are installed.

```
$ sudo yum install gcc make perlperl-modules libx11-dev libxext-dev libgl1-mesa-dev
```

2. Install the benchmark using wget command.

```
$ wget https://github.com/kdlucas/byte-unixbench/archive/master.zip
```

3. The next step is to configure the benchmark

```
$ make all
```

4. Execute the benchmark by Run script in top level directory. Run command compiles and execute the benchmark using its default makefile.

5. Single-threaded performance- Run the benchmark with c flag to specify one copy of test program run at a time.

6. `$/Run -c 1`

```

harika2903@test-1:~/byte-unixbench-master/UnixBench - Google Chrome
https://ssh.cloud.google.com/projects/main-ember-148813/zones/us-east1-b/instances/test-1?authuser=0&hl=en_US&projectNumber=94916991485
Benchmark Run: Sun Dec 11 2016 23:02:43 - 23:33:58
1 CPU in system; running 1 parallel copy of tests

Dhrystone 2 using register variables      13325424.3 lps (10.0 s, 7 samples)
Double-Precision Whetstone               897.8 MWIPS (27.5 s, 7 samples)
Exec1 Throughput                          1004.4 lps (29.7 s, 2 samples)
File Copy 1024 bufsize 2000 maxblocks    232744.6 KBps (30.0 s, 2 samples)
File Copy 256 bufsize 500 maxblocks       67600.2 KBps (30.0 s, 2 samples)
File Copy 4096 bufsize 8000 maxblocks     416575.7 KBps (30.0 s, 2 samples)
Pipe Throughput                           408041.2 lps (10.0 s, 7 samples)
Pipe-based Context Switching              92731.6 lps (10.0 s, 7 samples)
Process Creation                          3596.8 lps (30.1 s, 2 samples)
Shell Scripts (1 concurrent)              1012.3 lpm (60.0 s, 2 samples)
Shell Scripts (8 concurrent)              138.0 lpm (60.1 s, 2 samples)
System Call Overhead                      718929.1 lps (10.1 s, 7 samples)

System Benchmarks Index Values
Dhrystone 2 using register variables      116700.0  13325424.3  1141.9
Double-Precision Whetstone               55.0      897.8      163.2
Exec1 Throughput                          43.0      1004.4     233.6
File Copy 1024 bufsize 2000 maxblocks    3960.0     232744.6  587.7
File Copy 256 bufsize 500 maxblocks      1655.0     67600.2   408.5
File Copy 4096 bufsize 8000 maxblocks    5800.0     416575.7  718.2
Pipe Throughput                          12440.0    408041.2  328.0
Pipe-based Context Switching              4000.0     92731.6   231.8
Process Creation                          126.0      3596.8    285.5
Shell Scripts (1 concurrent)              42.4       1012.3    238.8
Shell Scripts (8 concurrent)              6.0        138.0     230.0
System Call Overhead                      15000.0    718929.1  479.3

System Benchmarks Index Score
=====
357.0

[harika2903@test-1 UnixBench]$

```

Figure-1: Sample Output showing the Single-Threaded Score

- Multi-threaded performance- Run the benchmark with c flag to specify eight parallel copies of test.

```

harika2903@test-1:~/byte-unixbench-master/UnixBench - Google Chrome
https://ssh.cloud.google.com/projects/main-ember-148813/zones/us-east1-b/instances/test-1?authuser=0&hl=en_US&projectNumber=94916991485
Benchmark Run: Sun Dec 11 2016 23:44:24 - 00:14:24
1 CPU in system; running 8 parallel copies of tests

Dhrystone 2 using register variables      15214149.1 lps (10.1 s, 7 samples)
Double-Precision Whetstone               1893.7 MWIPS (9.2 s, 7 samples)
Exec1 Throughput                          864.3 lps (29.6 s, 2 samples)
File Copy 1024 bufsize 2000 maxblocks    176577.7 KBps (30.0 s, 2 samples)
File Copy 256 bufsize 500 maxblocks       49305.9 KBps (30.0 s, 2 samples)
File Copy 4096 bufsize 8000 maxblocks     382251.5 KBps (30.0 s, 2 samples)
Pipe Throughput                           420120.6 lps (10.3 s, 7 samples)
Pipe-based Context Switching              85251.7 lps (10.1 s, 7 samples)
Process Creation                          3462.6 lps (30.1 s, 2 samples)
Shell Scripts (1 concurrent)              1046.3 lpm (60.3 s, 2 samples)
Shell Scripts (8 concurrent)              141.0 lpm (60.8 s, 2 samples)
System Call Overhead                      641787.5 lps (10.4 s, 7 samples)

System Benchmarks Index Values
Dhrystone 2 using register variables      116700.0  15214149.1  1303.7
Double-Precision Whetstone               55.0      1893.7     344.3
Exec1 Throughput                          43.0      864.3     201.0
File Copy 1024 bufsize 2000 maxblocks    3960.0     176577.7  445.9
File Copy 256 bufsize 500 maxblocks      1655.0     49305.9   297.9
File Copy 4096 bufsize 8000 maxblocks    5800.0     382251.5  659.1
Pipe Throughput                          12440.0    420120.6  337.7
Pipe-based Context Switching              4000.0     85251.7   213.1
Process Creation                          126.0      3462.6    274.8
Shell Scripts (1 concurrent)              42.4       1046.3    246.8
Shell Scripts (8 concurrent)              6.0        141.0     235.1
System Call Overhead                      15000.0    641787.5  427.9

System Benchmarks Index Score
=====
354.0

[harika2903@test-1 UnixBench]$

```

Figure-2: Sample Output for Multi-Threaded Score

Note: Unixbench 5.1.3 can also be downloaded from the below URL:

<https://github.com/kdlucas/byte-unixbench.git>

## APPENDIX B

### Dbench Installation and Execution

Launch the instance of same category (general purpose instance, memory-intensive instance, and compute-intensive instance) on Amazon EC2, Google Compute Engine, Microsoft Azure, and Rackspace. Use PuTTY or any other SSH client to connect to the public DNS of the instance.

The process for configuring and executing Dbench is described in following steps:

1. Before installing the benchmark, dependency packages are installed

```
$sudo yum install php-mysqlphpphp-xml php-mcryptphp-mbstringphp-cli  
mysqlhttpd
```

2. Install the benchmark using wget command.

```
$wget https://github.com/sahlberg/dbench/archive/master.zip
```

3. The next step is to configure the benchmark

```
$ ./autogen.sh
```

```
$ ./configure
```

4. Execute the benchmark using following command

```
$/dbench -B fileio -c loadfiles/client.txt -D ./ -t 30 <Number of Client processes>
```

- -B specifies the backend i.e. fileio.
- -c specifies the full path name of the client.txt file

- `-D <string>` specifies current directory in which dbench uses as the root i.e. `"/"`
  - `-t TIME`: benchmark is run for 30 seconds
  - `<Number of Client processes>` takes the number representing the number of client parallel processes.
5. Standard file I/O tests are run by varying the number of client processes- 1, 2, 25, 50,100,200, and 300.
- Example: Command to run 100 number of parallel clients simultaneously.
  - `$/dbench -B fileio -cloadfiles/client.txt -D ./ -t 30 100`

```

harika2903@instance-1:~/dbench-master - Google Chrome
https://ssh.cloud.google.com/projects/main-ember-148813/zones/us-central1-b/instances/instance-1?authuser=0&hl=en_US&projectNumber=945
dbench-master/clients/client25/~dmtmp/WORD/
dbench-master/clients/client25/~dmtmp/WORD/CHAP10.DOC
[harika2903@instance-1 dbench-master]$ ./dbench -B fileio -c loadfiles/client.txt -D ./ -t 10 100
dbench version 4.00 - Copyright Andrew Tridgell 1999-2004

Running for 10 seconds with load 'loadfiles/client.txt' and minimum warmup 2 secs
Waiting for child processes to finish setup.
Releasing clients
100      44      171.19 MB/sec      warmup      1 sec      latency 217.364 ms
100      80      89.51 MB/sec      execute      1 sec      latency 200.027 ms
100      95      82.30 MB/sec      execute      2 sec      latency 182.886 ms
100     108      79.32 MB/sec      execute      3 sec      latency 183.038 ms
100     131      77.76 MB/sec      execute      4 sec      latency 183.127 ms
100     155      76.83 MB/sec      execute      5 sec      latency 181.884 ms
100     177      75.26 MB/sec      execute      6 sec      latency 297.516 ms
100     204      76.21 MB/sec      execute      7 sec      latency 588.866 ms
100     227      75.81 MB/sec      execute      8 sec      latency 341.419 ms
100     247      74.78 MB/sec      execute      9 sec      latency 473.750 ms
100      cleanup      10 sec
0      cleanup      10 sec

Operation          Count      AvgLat      MaxLat
-----
Close              2575      13.137      271.569
WriteX             13627      65.671      269.943
QUERY_FILE_INFORMATION  2103      0.001      0.014
NTCreateX          2549      35.532      632.192

Throughput 74.7803 MB/sec 100 clients 100 procs max_latency=588.866 ms
[harika2903@instance-1 dbench-master]$

```

Figure 3: Sample Output for Standard File I/O Performance

6. File I/O performance overhead tests were performed by specifying the file I/O options with varying number of client processes-1,2,25,50,100,200, and 300.

- -F: When specified, Dbench performs fsync() to the file after each write operation. It blocks until all the buffered writes are complete.
- -s: This flag is passed to open() system call, indicating that all I/O on the file should be synchronized. When this flag is set, it uses O\_SYNC where each call to write() blocks until the data and metadata has been written to the disk.
- -S: Dbench calls fsync() function for all the directory operation-unlink, rename, rmdir, and mkdir operations.

Example: Command to perform file synchronous operations to the disk by hundred client processes.

`$. /dbench -B fileio -c loadfiles/client.txt -D ./ -t 30 100 -F -S -s`

```

harika2903@instance-1:~/dbench-master - Google Chrome
https://ssh.cloud.google.com/projects/main-ember-148813/zones/us-central1-b/instances/instance-1?authuser=0&hl=en_US&projectNumber=94916991485
[harika2903@instance-1 dbench-master]$ ./dbench -B fileio -c loadfiles/client.txt -D ./ -t 10 100 -F -S -s
dbench version 4.00 - Copyright Andrew Tridgell 1999-2004

Running for 10 seconds with load 'loadfiles/client.txt' and minimum warmup 2 secs
Waiting for child processes to finish setup.
Releasing clients
100 24 66.84 MB/sec warmup 1 sec latency 177.938 ms
100 53 75.07 MB/sec execute 1 sec latency 130.403 ms
100 65 75.28 MB/sec execute 2 sec latency 177.866 ms
100 80 74.21 MB/sec execute 3 sec latency 100.109 ms
100 95 73.68 MB/sec execute 4 sec latency 100.073 ms
100 106 73.36 MB/sec execute 5 sec latency 98.119 ms
100 131 73.49 MB/sec execute 6 sec latency 103.284 ms
100 153 72.87 MB/sec execute 7 sec latency 103.318 ms
100 177 72.70 MB/sec execute 8 sec latency 100.888 ms
100 202 72.57 MB/sec execute 9 sec latency 105.774 ms
100 cleanup 10 sec
0 cleanup 10 sec

Operation          Count    AvgLat    MaxLat
-----
Close              2062      0.004      0.029
WriteX            12798     79.427    177.860
QUERY_FILE_INFORMATION 1613      0.003      0.017
NTCreateX          2062      0.527     34.958

Throughput 72.5715 MB/sec (sync open) (sync dirs) 100 clients 100 procs max_latency=177.866 ms
[harika2903@instance-1 dbench-master]$

```

Figure 4: Sample Output for File I/O Performance Overhead

Note: Dbench can also be downloaded from the below URL

<https://github.com/sahlberg/dbench.git>

## APPENDIX C

### Iperf Installation and Execution

Launch the two instances (Client and Server) of same category (general purpose instance, memory-intensive instance, and compute-intensive instance) on Amazon EC2, Google Compute Engine, Microsoft Azure, and Rackspace. Use PuTTY or any other SSH client to connect to the public DNS of the instance.

The process for configuring and executing Iperf on Client and Server is described in following steps:

1. Before installing the benchmark, install development tools

```
$sudo yum groupinstall "Development tools"
```

2. Install the benchmark using wget command.

```
$wget https://github.com/esnet/iperf/archive/master.zip
```

3. The next step is to configure the benchmark

```
$/configure
```

```
$ sudo make
```

4. Execute the following command on Server instance to run Iperf in server mode.

```
$/iperf3 -s
```

5. Execute the following command on Client instance to run Iperf in client mode.

```
$/iperf3 -c <Private IP address of Server> -f k -t 1800 -i 60 -P <Number of  
concurrent client connections>
```



- -f m: shows the results in Megabits per second.
  - -t 1800: Run the benchmark test for 1800 seconds.
  - -i 60: interval in 60 seconds to report periodic bandwidth reports.
  - -P <Number of concurrent client connections>: Enables the number of simultaneous connections to make to the server. Default=1
6. Load testing on network is performed by varying the number of concurrent client connections- 1, 50, and 128. Maximum number of parallel connections allowed by Iperf is 128.

Example: Run the benchmark for ten seconds with two parallel connections from Client to Server.

`$./iperf3 -c 172.30.0.103 -fk -t 10 -P2`



Figure 5: Screenshot showing the Server Session.

```

harika2903@instance-2:~/iperf/iperf-master/src - Google Chrome
https://ssh.cloud.google.com/projects/main-ember-148813/zones/us-central1-b/instances/instance-2?authuser=0&hl=en_US&project=
[harika2903@instance-2 src]$ ./iperf3 -c 10.128.0.2 -f k -t 10
Connecting to host 10.128.0.2, port 5201
[ 4] local 10.128.0.3 port 38486 connected to 10.128.0.2 port 5201
[ ID] Interval      Transfer    Bandwidth   Retr  Cwnd
[ 4]  0.00-1.00  sec     208 MBytes 1747938 Kbits/sec    1   3.01 MBytes
[ 4]  1.00-2.00  sec     234 MBytes1960570 Kbits/sec   931   1.57 MBytes
[ 4]  2.00-3.00  sec     235 MBytes1974791 Kbits/sec    0   1.67 MBytes
[ 4]  3.00-4.00  sec     235 MBytes1970176 Kbits/sec    2   1.76 MBytes
[ 4]  4.00-5.00  sec     235 MBytes1975590 Kbits/sec   256   1.34 MBytes
[ 4]  5.00-6.00  sec     236 MBytes1978651 Kbits/sec    2   1.46 MBytes
[ 4]  6.00-7.00  sec     235 MBytes1975364 Kbits/sec    1   1.57 MBytes
[ 4]  7.00-8.00  sec     236 MBytes1977400 Kbits/sec    1   1.66 MBytes
[ 4]  8.00-9.00  sec     237 MBytes1984930 Kbits/sec    1   1.76 MBytes
[ 4]  9.00-10.00 sec     236 MBytes1976530 Kbits/sec    3   1.84 MBytes
-----
[ ID] Interval      Transfer    Bandwidth   Retr
[ 4]  0.00-10.00  sec     2.27 GBytes1952186 Kbits/sec  1198
[ 4]  0.00-10.00  sec     2.27 GBytes1949278 Kbits/sec
sender
receiver

iperf Done.
[harika2903@instance-2 src]$

```

Figure 6: Sample Output showing Networking Bandwidth

Note: Iperf 3.0 can also be downloaded from the below URL

<https://github.com/esnet/iperf>

## APPENDIX D

### Statistical Analysis

#### 1. Server Performance

Below are the listed tables for P-values of Unixbench tests on General purpose Instance, Memory-intensive Instance, and Compute-intensive Instance respectively. P-values were calculated for each pair of IaaS cloud platforms on the scores captured from four test sessions.

- General purpose Instance

Pair-wise list of IaaS platforms	P-values	
	Single-threaded test	Multi-threaded test
GCE Vs Amazon EC2	0.017	2.418E-09
GCE Vs Rackspace	0.03	2.516E-02
GCE Vs Azure	7.94E-07	7.08E-07
Amazon EC2 Vs Rackspace	0.289	2.55E-04
Amazon EC2 Vs Azure	0.001	1.473E-04
Rackspace Vs Azure	0.003	3.780E-04

Table 1: P-values for Unixbench tests on General purpose Instance

- Memory-intensive Instance

Pair-wise list of IaaS platforms	P-values	
	Single-threaded test	Multi-threaded test
GCE Vs Amazon EC2	0.007	0.012
GCE Vs Rackspace	0.774	0.287
GCE Vs Azure	0.825	0.143
Amazon EC2 Vs Rackspace	0.117	0.805
Amazon EC2 Vs Azure	0.370	0.090
Rackspace Vs Azure	0.756	0.093

Table 2: P-values of Unixbench tests on Memory-intensive Instance

- Compute-intensive Instance

Pair-wise list of IaaS platforms	P-values	
	Single-threaded test	Multi-threaded test
GCE Vs Amazon EC2	0.002	0.001
GCE Vs Rackspace	0.529	0.003
GCE Vs Azure	0.987	0.199
Amazon EC2 Vs Rackspace	0.008	0.001
Amazon EC2 Vs Azure	0.341	0.836
Rackspace Vs Azure	0.893	0.037

Table 3: P-values of Unixbench tests on Compute-intensive Instance

## 2. File I/O Performance

Below are the listed tables for P-values of Dbench tests on General purpose Instance, Memory-intensive Instance, and Compute-intensive Instance respectively. Two independent tests - standard file I/O test and file I/O performance overhead were performed on each IaaS platform across four test sessions. For each pair of IaaS platforms, P-values were calculated on the throughput results captured from passing different number of clients - 50, 100, 200, and 300.

- General purpose Instance

Pair-wise list of IaaS platforms	Number of Clients	P-values	
		Single-threaded test	Multi-threaded test
GCE Vs Amazon EC2	50	0.709333033	0.039126381
	100	0.262423866	0.488571476
	200	0.026569307	0.65870741
	300	8.40046E-10	0.463205376
GCE Vs Rackspace	50	0.000267523	7.46381E-05
	100	0.000153266	0.000280289
	200	0.00039946	0.000240383
	300	0.000461908	0.000385785
GCE Vs Azure	50	0.000997446	2.03315E-06
	100	0.000639842	9.65336E-09
	200	0.005346043	3.32488E-07
	300	0.009894774	1.06525E-06
Amazon EC2 Vs Rackspace	50	0.00021288	0.001753659
	100	4.14343E-06	0.006871719
	200	0.000385071	0.000659733
	300	0.000393591	4.77493E-05
Amazon EC2 Vs Azure	50	0.014370299	0.00171538
	100	0.011672423	0.049717599
	200	0.006251193	0.152352176
	300	0.021906741	0.120216884
Rackspace Vs Azure	50	1.76795E-06	3.85266E-05
	100	4.05373E-05	0.00010659
	200	6.58109E-05	0.000120257
	300	1.09477E-05	0.000213745

Table 4: P-values of File I/O (STD and OHD) Tests on General purpose Instance

- Memory-intensive Instance

Pair-wise list of IaaS platforms	Number of Clients	P-values	
		Single-threaded test	Multi-threaded test
GCE Vs Amazon EC2	50	7.03101E-08	1.23884E-05
	100	1.59391E-05	3.06473E-06
	200	0.000586051	6.77276E-09
	300	7.28084E-05	0.000151592
GCE Vs Rackspace	50	6.84863E-06	1.04837E-07
	100	5.78861E-06	1.12824E-07
	200	6.47089E-10	6.00337E-08
	300	1.65543E-06	2.64186E-07
GCE Vs Azure	50	3.71503E-09	2.66088E-05
	100	7.13141E-06	4.20468E-08
	200	4.50273E-05	4.94511E-08
	300	4.30131E-06	3.79049E-10
Amazon EC2 Vs Rackspace	50	1.71916E-05	0.000267537
	100	8.94896E-06	1.8597E-10
	200	1.22486E-09	1.90953E-07
	300	3.03776E-07	1.0596E-06
Amazon EC2 Vs Azure	50	2.59053E-10	4.80724E-09
	100	9.92834E-07	4.94284E-09
	200	3.74644E-06	8.84877E-08
	300	1.03983E-07	5.24751E-08
Rackspace Vs Azure	50	1.82315E-06	4.79561E-06
	100	7.79818E-06	1.58133E-10
	200	1.96341E-08	3.02301E-10
	300	9.43347E-08	2.52307E-07

Table 5: P-values of File I/O (STD and OHD) Tests on Memory-intensive Instance

- Compute-intensive Instance

Pair-wise list of IaaS platforms	Number of Clients	P-values	
		Single-threaded test	Multi-threaded test
GCE Vs Amazon EC2	50	0.194354224	0.039126381
	100	0.616087773	0.488571476
	200	0.03413113	0.65870741
	300	0.634807336	0.463205376
GCE Vs Rackspace	50	0.00084048	7.46381E-05
	100	0.00033031	0.000280289
	200	6.57692E-06	0.000240383
	300	0.000252123	0.000385785
GCE Vs Azure	50	2.00218E-07	2.03315E-06
	100	1.56781E-06	9.65336E-09
	200	4.385E-06	3.32488E-07
	300	6.58354E-05	1.06525E-06
Amazon EC2 Vs Rackspace	50	0.001637569	0.001753659
	100	1.18242E-05	0.006871719
	200	3.31933E-08	0.000659733
	300	7.60795E-06	4.77493E-05
Amazon EC2 Vs Azure	50	0.017428006	0.00171538
	100	0.053734758	0.049717599
	200	0.000987938	0.152352176
	300	0.076143256	0.120216884
Rackspace Vs Azure	50	0.000357928	3.85266E-05
	100	0.000223802	0.00010659
	200	3.59186E-06	0.000120257
	300	0.00021253	0.000213745

Table 6: P-values of File I/O (STD and OHD) Tests on Compute-intensive Instance



### 3. Network Performance

Below are the listed tables for P-values of Iperf tests on General purpose Instance, Memory-intensive Instance, and Compute-intensive Instance respectively. For each pair of IaaS platforms, P-values were calculated on the throughput results with different set of client parameters (50 and 128) captured from four test sessions.

- General purpose Instance

Pair-wise list of IaaS platforms	Number of Clients	P-values
GCE Vs Amazon EC2	50	2.13478E-10
	128	1.96982E-09
GCE Vs Rackspace	50	2.23463E-10
	128	3.01713E-08
GCE Vs Azure	50	2.79436E-07
	128	2.0859E-08
Amazon EC2 Vs Rackspace	50	8.98327E-12
	128	0.329754905
Amazon EC2 Vs Azure	50	1.78377E-05
	128	2.25325E-07
Rackspace Vs Azure	50	2.36748E-05
	128	5.40085E-05

Table 7: P-values for Network Throughput on General purpose Instance

- Memory-intensive Instance

Pair-wise list of IaaS platforms	Number of Clients	P-values
GCE Vs Amazon EC2	50	1.87855E-09
	128	2.08742E-07
GCE Vs Rackspace	50	4.21744E-05
	128	0.00959802
GCE Vs Azure	50	5.1363E-10
	128	2.56971E-09
Amazon EC2 Vs Rackspace	50	1.76086E-06
	128	0.00014703
Amazon EC2 Vs Azure	50	7.56253E-07
	128	2.51913E-05
Rackspace Vs Azure	50	1.64702E-06
	128	0.000405381

Table 8: P-values for Network Throughput on Memory-intensive Instance

- Compute-intensive Instance

Pair-wise list of IaaS platforms	Number of Clients	P-values
GCE Vs Amazon EC2	50	1.39363E-05
	128	5.85107E-05
GCE Vs Rackspace	50	1.82309E-06
	128	3.66657E-05
GCE Vs Azure	50	3.1336E-07
	128	8.80337E-06
Amazon EC2 Vs Rackspace	50	9.91215E-05
	128	1.44341E-06
Amazon EC2 Vs Azure	50	0.002326353
	128	0.018241124
Rackspace Vs Azure	50	0.000264555
	128	0.003391049

Table 9: P-values for Network Throughput on Compute-intensive Instance

## VITA

Niharika Deval earned her undergraduate degree in Computer Science from Jawaharlal Nehru Technological University (JNTU), Hyderabad, India. She is currently pursuing a master's degree in Computing and Information Sciences at the University of North Florida. She expects to receive her Master of Science in Computing and Information Sciences degree in the Spring 2017. She worked as Research Assistant for Dr. Sanjay P. Ahuja at University of North Florida, School of Computing, from Fall 2013 to Spring 2015.

### Journal Publications

Ahuja, Sanjay, P., and Deval, N., *System Level Benchmarks for the Cloud*, Computer and Information Science (CIS) Journal, volume 8, No. 2, May 2015.

<http://www.ccsenet.org/journal/index.php/cis/issue/view/1310>