Master Thesis

July 8, 2015

# Performance Analysis of Virtual Machines from Two Major IaaS Providers

## Amazon Web Services (EC2) and Google Compute Engine (GCE)

**Ilia Ryzhov**

of Chelyabinsk, Russia (12-722-609)

**supervised by**

Prof. Dr. Harald C. Gall

Dr. Philipp Leitner

University of Zurich[UZH]

s.e.a.l.

software evolution & architecture lab

Master Thesis

# Performance Analysis of Virtual Machines from Two Major IaaS Providers

## Amazon Web Services (EC2) and Google Compute Engine (GCE)

**Ilia Ryzhov**

University of Zurich[UZH]

s.e.a.l.
software evolution & architecture lab

**Master Thesis**

**Author:** Ilia Ryzhov, ilia.ryzhov@uzh.ch
**URL:** www.iliaryzhov.com
**Project period:** 19th Jan 2015 - 9th Jul 2015

Software Evolution & Architecture Lab
Department of Informatics, University of Zurich

# Acknowledgements

I would like to thank Dr. Philipp Leitner for his immense support in the thesis process and for providing financial means to work with cloud services.

# Abstract

Cloud computing is an ongoing trend in Internet computing. In cloud computing, resources, such as CPU processing time, disk space, or networking capabilities, are rented and released as a service. Today, the most important model for delivering on the cloud promise is the Infrastructure-as-a-Service (IaaS) model. In IaaS, virtual computing resources (most importantly virtual machines) are acquired and released on demand, either via a Web interface or programmatically via an Application Programming Interface (API).

As IaaS is receiving a significant hype in industry, a large number of commercial vendors have started to appear, providing IaaS services (e.g., Amazon's EC2 - AWS, Google's Compute Engine - GCE, Microsoft's Azure, or Rackspace's Public Cloud Hosting). Functionally, these services are largely equivalent, however, the non-functional properties (performance, reliability, costs, etc.) vary significantly. In order to support software engineers in selecting the best cloud service for their applications, many researchers have started initiatives to evaluate the performance of these services.

This work performs a deep comparison and analysis of two major IaaS service providers - AWS and GCE. A number of benchmarks had been run on different virtual machine (instance) types of both providers, analysis performed, and according to the benchmarks' results and costs of running a virtual machine, recommendations were given, which machines from which provider suit certain task better.

# Zusammenfassung

Cloud Computing ist ein fortlaufender Trend im Internet Computing. Die Cloud Computing-Ressourcen wie die CPU-Verarbeitungszeit, Arbeitsspeicher oder Netzwerkfähigkeiten, werden vermietet und als Dienst freigegeben.

Das Infrastructure-as-a-Service (IaaS) Paradigma ist heute das wichtigste Modell für die Umsetzung des Cloud-Versprechens. Im IaaS werden virtuelle IT-Ressourcen (vor allem virtuelle Maschinen) erfasst und bei Bedarf freigesetzt. Die Freisetzung geschieht entweder über eine Web-Schnittstelle oder über die programmgesteuerte Application Programming Interface (API).

IaaS geniesst grosses Ansehen in der Industrie. Viele Anbieter wie Amazon EC2 (AWS), Googles Compute Engine (GCE), Microsofts Azure oder Rackspace Public Cloud-Hosting bieten IaaS Dienste an. Funktionell entsprechen sich diese Dienste weitgehend, jedoch variieren sie erheblich mit nicht-funktionalen Eigenschaften (Leistung, Zuverlässigkeit, Kosten, etc.).

Um Software-Ingenieure in der Auswahl des besten Cloud-Services für ihre Anwendungen zu unterstützen, haben Forscher Initiativen gestartet, um die Leistungserbringung von IaaS zu bewerten und zu verbessern. Diese Arbeit setzt sich mit dem Vergleich sowie der Analyse von zwei grossen IaaS-Dienstleister, AWS und GCE, auseinander. Eine Reihe von Benchmarks hat auf verschiedenen virtuellen Maschinen die Arten von beiden Anbietern aufgeführt und anschliessend Untersuchungen durchgeführt. Aufgrund der ermittelten Ergebnisse sowie der eruierten Kosten wurden Empfehlungen abgegeben, welche virtuelle Maschinen aufgrund der ermittelten Faktoren besser zum jeweiligen Anbieter passen.

# Contents

# List of Figures



# List of Tables

# Chapter 1

# Introduction

Many, if not all, businesses today use different web applications in their activities. It can be an online store, a blog, file hosting or sharing service, or just a website. In order to allow large number of customers to use them concurrently, these applications have to be reliable and able to serve all users without any significant delays or outages. At the planning stage it is difficult to determine how large the audience will be, and business owners need to find a compromise between the server performance and a price.

In standard businesses, where all IT infrastructure is located on site, changing the size of computing powers and storage capacity is a time consuming and expensive activity, which requires a number of specialists to be involved. If this process lasts too long, it leads to customer dissatisfaction. There is also a risk of money overspending if the wrong hardware specifications had been chosen. Moreover, the costs of maintaining, servicing and updating this infrastructure is a full responsibility of the business owner.

Another drawback of the traditional way of hosting IT infrastructure on site is low scalability and slow adoption time to new emerging requirements. Once the new idea or a project had been confirmed by management, it takes relatively long time to buy and configure new servers.

The invention of cloud services completely changed the way how companies can keep and manage their IT resources. Modern cloud service providers offer computing services, storage, platforms that help to build and run applications in cloud. Clouds provide a possibility of quick provisioning of IT resources without significant financial investments and long configuration time. Most cloud services today use "pay-as-you-go" billing system, when customers only pay for resources that they used. If no resources are in use at the moment, users are not charged.

This fact is one of the key differences between cloud services and traditional computing centers. If you want to have your own computing center, you first have to spend money on hardware, then you need to configure all infrastructure, and pay for maintenance and electricity. Moreover, you need to have knowledge of how to build such an infrastructure. If you use cloud services, you can gain access to virtual computing, storage, database or any other rented resources from a personal computer within minutes. Customers only need to specify what type of service they need and select desired performance based on the data provided on the website. No special knowledge is required.

Leading cloud service providers have infrastructure located on all continents, and cloud users around the world can benefit from that and rent services with similar performance at any location. Such geographical distribution provides powerful resources at low cost [76], [80].

Cloud services are especially beneficial for small companies, especially startups, or individuals, who wish to start projects and implement their ideas quickly and without significant investment, and be able to change the hardware at any time. Such users don't have a possibility to run servers on site, because they might not have enough space or money to pay for expensive hardware, or they just don't want to. This makes new ideas feasible and easy to implement.

But most cloud services, at least at the current level of development, have certain limitations that have to be considered before choosing a cloud service provider. A virtual machine, even though it appears to be as a physical system, is heavily dependent on the physical hardware that it got assigned to, and the virtualizer, that takes care of the process of assigning virtual machines to hardware components, located in the computing center [54]. The virtualization process causes virtual machines to have variable performance data even between machines of the same type, created at the same time, and it also leads to virtual machines having different performance after restart, or just at a different time of a day. These limitations of cloud services are among the reasons why clouds are not yet widely deployed at companies, and why most businesses rely on the infrastructure that they have on site [58], [83].

## 1.1 Motivation

The number of IaaS provides is growing every year, and today we have more than ten global vendors, which offer virtual instances or virtual machines for private and business use. Compared to the number of providers five years ago, now the choice of vendors and cloud services they offer is significantly larger. Such competition drives the prices down, forces providers to improve their services and to introduce additional features.

On the other hand, due to such variety the process of choosing the right cloud provider and services has become more complicated. Customers have to consider many factors when making this choice, including performance, reliability, stability, price and many others. Each of these factors is important and sometimes they are mutually exclusive [12], [59]. Often customers can get similar instance configurations from different providers, but the price will differ significantly, and the question arises whether the actual performance of the cheaper offering is the same as offered by another provider.

Findings of this research could support scientists, researchers and developers, who need to use cloud services to accomplish their goals. The experiments, performed in this work will provide customers of infrastructure-as-a-service (IaaS) cloud providers with the data that will help them to make decision about which providers is more suitable for particular task, and which provider offers better virtual machines for a cheaper price and performance of which virtual machines is generally higher.

This work compares performance of virtual machines of two providers from multiple perspectives, and the results and analysis can assist in deciding which offering suits particular task better.

## 1.2 Goal

The main task of this Thesis is to perform analysis and compare performance of different virtual machines of the most common types of two major IaaS providers - Amazon EC2 and Google GCE, since they are often referred to as leading IaaS providers [13]. A set of benchmarks has been carefully selected for the purpose of analyzing virtual machines from different perspectives: memory and processor performance, network bandwidth and latency. After the experimental data has been collected, the analysis will be performed to figure out, why a certain component has better performance at one cloud provider than at another. Amazon and Google deploy different hardware in their computing centers, and they use different virtualization techniques, and this definitely has its affect on the performance that virtual machines have.

The goal is also to provide recommendations on which particular type of virtual machine, offered by Amazon EC2 or Google GCE (one of the "general purpose" types, that include Small,

Medium, Large and Xlarge) is more suitable for certain tasks - most common tasks for which virtual machines can be used today. Also, price analysis in relation to the actual performance of similar offerings from Amazon and Google was made to give an overview of the price difference between these two cloud providers for similar types of VMs.

## 1.3 Research Questions

Based on the goals of this thesis, there are two research questions as follows. The first question relates to the correspondence between the characteristics of virtual instances specified on the provider's website and their real performance, estimated by the set of specialized benchmarking tools. Due to possible difference in hardware deployed in different data centers of a cloud provider, performance data of similar virtual machines can vary. Also, the performance of a virtual machine heavily depends on the virtualization process and on other virtual machines, that are assigned to the same physical hardware components in a data center. These and other factors affect the actual performance of VMs leased by customers, and the first research question of this work addresses this issue and asks whether there is a difference between the actual performance of virtual machines and the specifications, presented on the provider's website.

**RQ1**:
*Does the real performance of virtual machines match the characteristics specified by the service provider?*

The process of virtualization never runs in exactly the same way in different computing centers. Also, different service providers can use different virtualization techniques. Based on the virtualization algorithm, hardware components configuration and current utilisation level, virtualizer will choose different hardware to create a virtual machine on. And the second question asks whether instances with similar characteristics from Amazon EC2 and Google GCE can perform in the same task with similar results.

**RQ2:**
*Do similar virtual machines in Amazon and Google have similar performance?*

## 1.4 Structure of Thesis

The thesis has the following structure. In the chapter Background, some general information about Amazon EC2 and Google GCE providers is presented as well as the set of services and products they offer. In Related Work section, research that has been done before in the area of cloud service benchmarking and virtual machine performance analysis is described. Results of previous studies are relevant to this work, since some of the findings are used in this thesis.

Then, in chapter 4 - Study Setup, there is an overview and configuration details of the benchmarks and tools used in this work. Also, problems which the author confronted and had to overcome during the installation and configuration process are mentioned in this chapter. The benchmarking data and experimental results together with some explanation and discussion are outlined in chapter 5 - Results and Data Analysis.

Finally, in Discussion the most interesting findings and results are discussed and also there are some recommendations for researches and developers about which instance type from which service provider would be more suitable for certain tasks, taking into account performance data and VM rental prices.

# Chapter 2

# Background

## 2.1 Cloud Service Models

Most cloud providers offer different types of services. Customers can choose any product depending on their needs. It can be either a ready-to-use service, a platform for development and customization of services, or a virtual machine, which can be used for any purposes. These products are divided by groups according to following classification: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS).

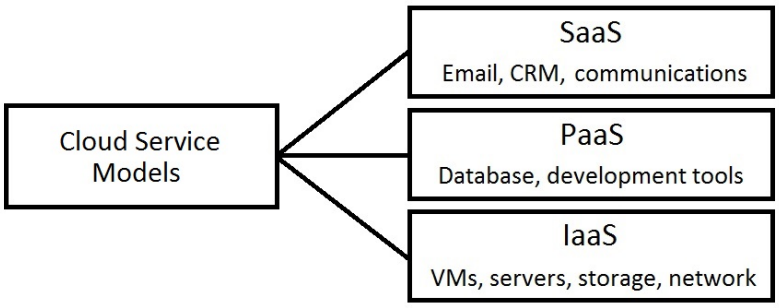


**Figure 2.1**: Cloud service models

The following figure demonstrates the dependencies between different cloud service models. SaaS is built upon PaaS, and PaaS is built upon IaaS. Also, main actors that deal with corresponding cloud service models are displayed.

Each cloud service model is described in more detail in following sections.

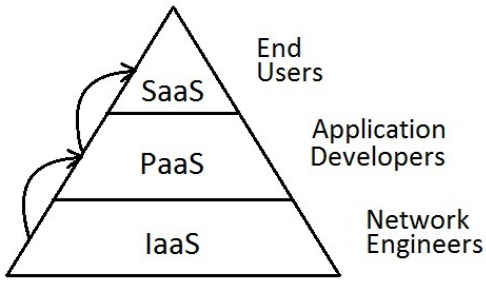


**Figure 2.2**: Dependencies between cloud service models

## Software as a Service (SaaS)

The most popular service model today is Software as a Service (SaaS), when provider offers a finished product, which can be used by customers without the need to develop or configure anything. Only some customization can be performed by users. Most people are familiar with this cloud service model and use it on a daily basis. Examples of SaaS are email clients like Gmail[1] or Yahoo[2], Customer relationship management software (CRM), Microsoft Office online[3], and others.

Users can use these applications without having to take care of installation, because service providers perform installation, configuration and updates, making sure that these services always work as intended.

In SaaS model users don't have administrative access to the underlying infrastructure. The application already has a default configuration, and users can perform some customization if desired. This allows a large number of customers to use that service, because they don't have to obtain any specific knowledge to be able to use it. As a rule, for large applications, a number of servers run the same application for scalability, redundancy and load balancing between different regions.

Many businesses use services based on a SaaS model. For example, email client, Microsoft Office online, CRM, etc. Such services allow simplifying the configuration part of the work that has to be done at a company, saving time and money. A drawback of SaaS is that all customer data is stored on a server side, which could be a potential security risk. Also, monthly or annual chargers apply to most of these services.

## Platform as a Service (PaaS)

The PaaS model is oriented on more advanced users, because it provides a platform and a set of tools for developing a broad range of applications. All infrastructure is provided, and users don't need to manage it while developing an app. PaaS can be provided in two different ways: as a public cloud service, where a customer can manage software deployment and configuration settings, and provider takes care of network communications, servers and storage; or as software,

[1] http://www.gmail.com
[2] http://www.yahoo.com
[3] http://office.live.com

that is installed in data centers on site and managed by a customer [14]. Administrative access to servers and databases in case of public cloud is restricted.

Customers need to have high level programming skills to develop applications in a PaaS service model. The scaling and load balancing can be done automatically.

One of the features of PaaS is that the underlying resources are managed automatically to match the user's app demands, so users don't have to worry about possible lack of CPU speed or RAM. Examples of such services are Microsoft Azure[4] and Google App Engine[5].

The disadvantage of PaaS is that applications, developed on a certain platform, cannot be easily moved to another platform [15]. For example if an app was built on Google Apps, it will be running only on this platform.

### Infrastructure as a Service (IaaS)

This cloud model is oriented on advanced users, server designers and developers. These categories of users can rent virtual machines with almost any specifications, install any operating system on them, and use these VMs for any purpose. Storage and network services are also available for rent.

On providers' side hypervisors are used to support and scale a large number of virtual machines running on physical hardware in data centers. Other resources, which are often used in combination with virtual machines, are operating system image library, block storage, file storage, firewalls, load balancers, IP addresses, virtual local area networks (VLANs), etc.

Customers are usually billed on a per-usage basis, depending on how long they run virtual machines for.

The disadvantage of IaaS model is that it is not always possible to predict the exact performance of a particular virtual machine, which will be assigned to a user [3]. The characteristics of a VM depend on virtualizer and physical hardware, where this virtual machine will be located, and on another cloud users, who can also get a virtual machine located on the same physical hardware [8].

The IaaS services are evaluated in this work. A number of virtual machines with different performance from Amazon EC2 and Google GCE had been created, evaluated and documented to answer the research questions of the thesis.

## 2.2 IaaS Services

There are many IaaS service providers on the market today. The most popular ones are Amazon Elastic Compute Cloud (EC2)[6], Google Compute Engine (GCE)[7], Rackspace Open Cloud[8], Windows Azure, IBM SmartCloud[9]. Different types of IaaS services and the parties that have control over respectful types are presented on the figure below.

[4]http://azure.microsoft.com
[5]http://appengine.google.com
[6]http://aws.amazon.com/ec2/
[7]http://cloud.google.com/compute/
[8]http://www.rackspace.com/cloud
[9]http://www.ibm.com/cloud-computing/us/en/private-cloud.html

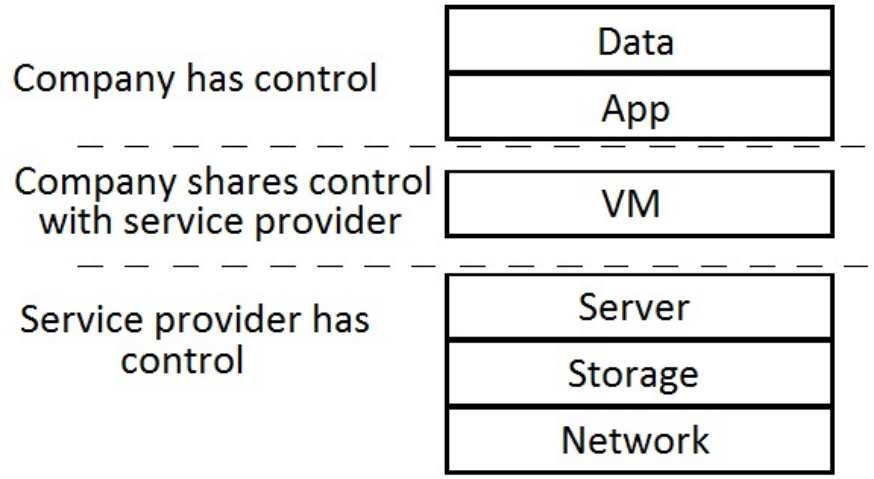


**Figure 2.3**: Types of IaaS services

Within this thesis a series of experiments had been conducted with virtual machines of two major cloud providers - Amazon EC2 and Google Compute Engine. This section provides a brief overview of these service providers.

### 2.2.1 Amazon EC2

#### Overview

Amazon Elastic Compute Cloud (Amazon EC2) cloud service provides virtual machines (called virtual instances) of different sizes (types), designed for different purposes. Customers can rent a virtual machine for any time, and they pay on the per-usage basis.

Rented virtual machines can be accesses via SSH or HTTPS protocols [18]. Amazon EC2 allows its customers to quickly start new virtual machines, discard the machines that they don't need anymore, create images of their virtual machines to recreate them or change the type without spending much time on configuration.

When starting a new virtual machine, Amzon EC2 users can choose out of many operating systems and machine templates available in the Amazon's database. There are templates with specialized configurations and software already installed and configured, for example LAMP image, that includes Apache, MySQL and PHP installed and configured for Linux OS.

All rented virtual machines come with dedicated disk space. By default, each machine gets 10 GB of disk space, but customers can choose alternative disk capacity options, which cost extra money. Also, by default customers get a single disk for each instance. But it is possible to make RAID structure, having two or more disks for redundancy.

Amazon classified its virtual instances by main parameters, and customers can find the virtual machine that will suit their particular needs. Most customers will choose one of the "General purpose" virtual instances, and more advanced users, developers and researches can start specialized instances, for example "High CPU", which has high performance multi-core processors with relatively small amount of RAM, or "High RAM" or one of many other types.

Besides computational power, Amazon also provides advanced network infrastructure. All Amazon EC2 virtual instance are located in one virtual private cloud (VPC), which is an isolated

network, similar to LAN. Customers can configure different permissions to access their instances, and they can also create access rules similar to access control lists (ACLs). Public IP address can be assigned to instances, allowing them to be accessed from outside of the Amazon EC2 network.

## History

Amazon first appeared on the market in 2006 with a limited number of virtual machine types and resources for lease. In 2008 it had more than ten types of virtual instances, including few "general purpose" and "high CPU" types, and enough resources to provide uninterrupted service for all customers [16]. Also, in 2008 new zones were introduced with computing centers located on different continents, possibility of assigning a static IP address and persistent storage [16]. At the same time Amazon announced its service-level agreement (SLA) and plans to build a management console, where users can manage rented virtual instances in a web browser This feature was implemented in 2009 [17].

## Current offerings

Amazon offers different price schemes that can suite different groups of customers[10]:

**On-Demand Instances.** On-Demand Instances is the traditional method of payment, when customers are charged on "pay-as-you-go" basis. This method doesn't require any upfront time and costs planning. This method suits the majority of users who doesn't have experience using Amazon EC2 products, and this method is used by default.

**Reserved Instances.** If customers know beforehand when, for how long and which instances they need, they can make a reservation of certain resources. In this case the price is lower, but this method requires thorough upfront planning. This option is suitable for scheduled processes, that happen periodically over time. There are three reserved payment options: "No upfront", "Partial upfront" and "All upfront", which differ by the sum of money that has to be paid in advance.

**Spot Instances.** Pretty often data center capacities are loaded much less than 100 per cent, also sometimes users cancel their reservations for virtual machines. Customers can bid on unused resources and get them for cheaper price if their bid wins. As a rule, the time when such capacities are available, is not convenient for users (late at night), but if the budget of a project is strictly limited, spot instances is a good choice.

In this work only On-Demand instances were used due to the fact that the goal was to study the instances, which are used by the largest amount of customers, and to use the same payment method. Since most customers use On-Demand payment method, the choice was obvious.

The following table displays prices (as of May 2015) for the most commonly used Amazon EC2 instances - type standard, or "general purpose" virtual instances (all running either version of Linux and located in region Western Europe). They are optimized by CPU processing speed and the amount of RAM. These instance types were also utilized in this work for benchmarking purposes.

[10] http://aws.amazon.com/ec2/pricing/

| Machine Type | Virtual CPUs | Memory, GB | Price per hour (USD) |
|---|---|---|---|
| t2.small | 1 | 1 | 0.014 |
| m3.medium | 2 | 3.75 | 0.077 |
| m3.large | 2 | 7.5 | 0.154 |
| m3.xlarge | 4 | 15 | 0.308 |
| m3.2xlarge | 8 | 30 | 0.616 |

**Table 2.1**: Amazon EC2 Pricing

### 2.2.2 Google Compute Engine

#### Overview

Google Compute Engine (GCE) is an IaaS cloud service similar to its competitor Amazon EC2. Google GCE provides its customers computational capacities on the "pay-as-you-go" basis. Resources are being leased in the form of virtual machines (instances). The main features of GCE are similar to those of Amazon EC2 and include: virtual machines of different types that suit different user needs, static and public IP addresses, load balancing techniques, grouping machines in clusters, etc. [20]

#### History

The first announcement of Google Compute Engine was in 2012 [19]. In 2013, GCE became available for general customers, and at that time it already had computing centers located on different continents. The development of GCE was happening very fast, and in the same year of 2013 Google announced some new features like shared core virtual instances, enhanced network performance, different billing methods. GCE attracted many customers in the first operational year by offering high performance at low prices. In 2014 such features as support of Windows Server OS, DNS services were added for virtual machines [22].

#### Current offerings

Google Compute Engine utilizes traditional billing approach, when customers are charged for the amount of resources they used over the time period. There are though some pricing plans, similar to Amazon EC2. The one interesting feature is that GCE charges extra money (around 30 per cent of the hour rate) if a virtual machine was loaded less than 25 per cent over the last month. On the contrary, the price gets lower if a VM was used the whole month without stops [23].

The table below represents current pricing (as of May 2015) for general purpose virtual machines in GCE (with Linux OS and located in region Western Europe). These types of VMs are used most often, because their CPU and RAM are optimized for most common user tasks. Instances of these types were used also in this work.

| Machine Type | Virtual CPUs | Memory, GB | Price per hour (USD) |
|---|---|---|---|
| n1-standard-1 | 1 | 3.75 | 0.042 |
| n1-standard-2 | 2 | 7.5 | 0.084 |
| n1-standard-4 | 4 | 15 | 0.168 |
| n1-standard-8 | 8 | 30 | 0.336 |
| n1-standard-16 | 16 | 60 | 0.672 |

**Table 2.2**: Google GCE Pricing

## 2.3 Cloud Benchmarking

When customers make decision about what cloud service to choose, they want to test the cloud performance, or to see results of tests done before; they are often looking to compare performance of different cloud providers. A simple price comparison may lead to choosing provider with poor performance. What really matters is combination of the price and performance that would suit the customer. Cloud services' users wish to know how much it costs to run a specific application or accomplish a certain task using infrastructure that is offered. Therefore a meaningful comparison should consider both the price and the cloud performance to present some measure of cost per computing unit [8], [76].

### 2.3.1 Goals of Cloud Benchmarking

Before performing any benchmarking it is important to decide which hardware component should be benchmarked. This will determine the relative importance of different performance factors [8]. For example, if a customer needs a database server and this server is expected to be under heavy load, then the disk I/O performance should be the key factor while considering different offers, and the user will be interested in benchmarks that evaluate I/O performance of a virtual machine.

The goal of cloud benchmarking is to find the best cloud service that suits certain customer needs. Most important performance factors, that are usually considered when choosing the service provider are CPU processing speed, amount of RAM, I/O speed and network performance. All these factors matter with a certain degree, unless some specific virtual machine is required, for example "High CPU", and when choosing the best offering for the "general purpose" virtual instance, the goal is to find a perfect combination of these parameters that will suit customer needs [24].

Before starting a benchmarking process, a user of cloud services needs to define what s/he will consider to be good performance of a cloud i.e. what aspects are key and have biggest impact on the real performance of cloud based on user's specific requirements. In case if a cloud user needs good overall performance, then s/he should draw his attention to those benchmarks that evaluate the performance of all hardware components and their interoperability. In case if s/he needs a virtual instance for some specific needs, as mentioned earlier, then micro benchmarks that test separate components should be used. Once these aspects are defined, then it's time to start looking at actual benchmarking process.

### 2.3.2 Factors to be Considered in Cloud Benchmarking

Cloud benchmarking process varies depending on what component of a system is being evaluated [25], [64], [66], [75].

### Computational Performance

One of the processes that almost often takes place in IaaS clouds is contention [26]. Contention happens because almost none of leased virtual machines are used at loads close to 100 per cent all the time. This is similar to flight overbooking by most airlines. The CPU allocation is controlled by virtualizers, and sometimes virtualizers can reallocate resources between different virtual machines. For users this process happens insensibly, but it can be tracked by special tools [54], [56].

Contention is the reason why in some clouds the actual performance can change sometimes, depending on the hardware which has been allocated for the virtual machine, and usually the degree of contention is unpredictable [56].

### Storage

In modern clouds storage often becomes a performance bottleneck [46], [47]. Virtualizers are capable of allocating CPU and RAM resources in such a way that overlapping or contention happens relatively seldom, and cloud users often don't realize that the CPU, that their virtual machine is associated to, might be shared with some other processes. But storage is the factor that limits the actual performance of a cloud service today. For high performance virtual instance, the disk can become the component that will limit the overall performance of that virtual machine [25]. Solid state drives (SSDs) can be a solution in a short perspective, because the technology used in CPU and RAM production evolves faster than the data storage technology.

Storage component can be the reason of a system failure more often than CPU or RAM, and the consequences of such a failure in most cases are more severe. But for critical tasks and systems, disks are always mirrored using RAID technology.

### Network

The network benchmarking is less complex than evaluation of CPU or RAM performance, and there are many tools available that can evaluate different network performance values, such as bandwidth, latency, jitter, delay, load time, etc. The two factors that play key role in cloud network benchmarking are bandwidth and Latency [72], [73].

Latency is a time between the request and response packets. Latency is critical for the tasks where real time communications are involved. But in case of virtual machines in cloud, latency is not critical, it is just one of the factors, that characterises the network performance [28]. High latency value means that two virtual machines are located physically far away from each other. That is the reason why all pairs of virtual instances used in this work were located in a single LAN, that belongs to one cloud service provider. Bandwidth is the amount of provided or consumed information, measured in bits per second. Cloud users usually pay for the network services based on the amount of bandwidth they used. Both latency and bandwidth are evaluated by downloading a sample file and measuring the download time and the delay between request and response [28], [72].

## 2.3.3 Tools for Cloud Benchmarking

There is a number of tools for cloud performance benchmarking available today. Most of them come from the area of server administration. There is no specialized tool set or a package which was specifically designed for evaluating cloud services, and therefore all these tools have to be adopted for the use in cloud infrastructure.

### Computational Performance

There is a large number of tools which can measure CPU and RAM performance of servers. Starting from a Linux system monitoring tool "top", which displays system summary information in real-time, to platforms that include multiple components aimed to simulate applications running on a system and measure the system's performance while executing certain tasks [71], [91]. However the most popular and easy-to-use applications for performance benchmarking are small tools, or micro benchmarks, that can easily be installed and run. Such tools simulate system processes for load testing CPU, RAM or any other component separately.

All three types of benchmarking tools described above were used in this work. The integrated platform RUBiS[11] was used to simulate an online auction website similar to eBay[12], to measure the overall performance of the virtual machine. Similar to RUBiS, but already obsolete benchmark, TPC-W[13], could have been used for simulating a transaction-based website. This benchmark simulates the functions of a web server and outputs a number of web interactions per second (WIPS), that can be processed by a server. A number of more lightweight tools like Phoronix Test Suite[14] were used to benchmark both CPU and RAM using different micro benchmarks. These and other tools used in this work are described in more detail in chapter Study Setup.

### Storage

The Phoronix Test Suite used in this work can be also perform storage performance evaluation, because it contains several micro benchmarks for that purpose. The factors that have to be taken into account when benchmarking storage are [25]:

- Storage architecture (local, SAN);
- Redundancy implementation (RAID);
- Temporary or permanent storage.

### Network

As mentioned before, the classical approach to estimate bandwidth and latency is to download a sample file and to measure the download time and the time between request and response. There are many tools that automate this process. It worth mentioning that when evaluating network performance, all measurements have to be done few times because of the processes like ARP request, MAC address caching and DNS lookup, that take place when the address is queried for the first time.

The tools used in this work for network benchmarking are JMeter[15] for measuring latency and page load time, and IPerf[16] for measuring bandwidth.

[11] http://rubis.ow2.org/
[12] http://www.eBay.com
[13] http://www.tpc.org/tpcw/
[14] http://www.phoronix-test-suite.com
[15] http://jmeter.apache.org
[16] http://iperf.fr

# Chapter 3

# Related Work

Cloud computing is a relatively new field of research and the first clouds in the form in which we know them, came into existence in 2000s. Clouds became widespread in the end of 2000s, when several providers started offering cloud products for lease to general customers. Since then researches and developers started exploring the new product, comparing providers, evaluating performance and writing research articles about their work. From the user's perspective the IaaS cloud is a virtual machine accessible via SSH, so the first cloud benchmarking was similar to traditional physical server's benchmarking, but later new methods for cloud performance were introduced, which are described in this chapter.

## 3.1 Analyzing the Functionality of Cloud-Based Virtual Machines

This section covers the related research that has been done in the field of investigation and analysis of IaaS clouds and their functionality. The majority of papers are dated back to the times when pioneers of IaaS services had only entered the market, and people were curious about this technology.

J. O'Loughlin and L. Gillam in the article "Towards Performance Prediction for Public Infrastructure Clouds: an EC2 Case Study" [10] raise the issue of difficult choice that people face when they need to select the right virtual machine for their needs. This paper uses statistical methods for performance prediction in IaaS clouds. The authors demonstrate the possibility of the chance to get a virtual machine with a certain performance. Based on the results they got, the authors suggested that the SLA and pricing should be adopted to reflect the performance differences.

S. Ostermann et al. in the article "A Performance Analysis of EC2 Cloud Computing Services for Scientific Computing" [5] address the problems of virtualization at computing centers. They study the ways how virtualization mechanisms can be implemented and their influence on the actual performance of leased virtual machines. The authors draw their attention to Amazon EC2 services and run a series of micro benchmarks to test their performance. The main finding is that the current level of performance predictability is low. They specify, that virtualiziers cause similar VMs to have different performance. Based on their results, the authors concluded that at the time of writing (2010) the IaaS clouds were not suitable for scientific computing.

J. Dejun et al. in "EC2 Performance Analysis for Resource Provisioning of Service-Oriented Applications" [8] article again discover the problem of performance variations on similar instances. The authors consider instances of type Small from Amazon EC2. They state that machines have stable performance, but it's not possible to predict what performance the instance will have. Based on the results they got, the authors suggest using virtual machines, that should

have had the same performance, for different purposes, depending on their actual performance. For example, if a VM has high I/O performance data, it can serve as a database server. This paper was written in 2010, and now, five years later, the resource provisioning at Amazon has significantly improved, and this fact has been confirmed by the results of this work.

P.Leitner and J. Cito in the article "Patterns in the Chaos-a Study of Performance Variation and Predictability in Public IaaS Clouds" [1] present a large-scale literature review about the research done in performance evaluation and predictability in IaaS cloud services. They stated fifteen hypotheses, which were validated by the experiment. Based on their findings, the authors concluded that Google GCE virtual machines are in general more predictable than Amazon EC2 VMs.

Craig Lee in his paper "A perspective on scientific cloud computing" [74] points out that the cloud technology will soon replace the traditional way of scientific computing, and that clouds will also become more popular among general customers, who use cloud-based virtual machines for their private needs.

## 3.2 Benchmarking of IaaS Cloud Services

The second section in this chapter addresses the research that has been done in the area of IaaS cloud benchmarking. The main research activities in this field start after 2010, when several cloud providers have entered the market, and the general quality level had significantly improved since previous decade. Most experiments, described in these articles presented in this section, are closely related to experiments conducted in this work.

Much work has been done since 2010 on investigating the cloud services performance, benchmarking of virtual machines and analyzing their performance. There are many scientific papers [60], [64], [66], [67], [71], [75], [77], [86], [90], [92] and [93], which perform benchmarking of cloud-based virtual machines using different set of benchmarks, including those used in this work. The article [49] uses statistical methods for performance evaluation. As papers have been written at different times, the results of experiments, presented by the authors are also different. Moreover, these authors have been conducting their experiments with cloud services in different locations on the Earth, and therefore the results of similar services differ. This set of articles gives on overview of the general procedure of cloud benchmarking, presenting different approaches that the authors used in their experiments.

A. Lenk et al. in the paper "What What are you paying for? performance benchmarking for infrastructure-as-a-service offerings" [3] raised the issue that the performance specified on providers' websites doesn't reflect the actual performance of leased virtual instance. This research considers Amazon EC2, Rackspace and Flexiscale virtual machines. The authors proposed a benchmark suite that would contain all necessary tools for cloud benchmarking already configured in a necessary way to suit benchmarking of cloud services. As they mentioned, this would decrease the amount of time and effort required to estimate the real cloud performance. Another their suggestion was creation of virtual machines, equipped with a benchmarking software, that can be used to measure and compare performance between different providers. There are few more articles that also compare the actual cloud performance and the price that customers have to pay for rented VMs: [52], [76], [80]. These authors perform performance benchmarking in combination with costs analysis, and conclude which offering would suit a certain task better. All these papers have a limitation, that prices for cloud products constantly change, and the hardware used in data centers also change. So these articles are only valid while all these conditions stay the same as at the time of experiment. But readers can get the overview of the experiments conducted and the methods for costs analysis in order to perform similar research, but with the new virtual machines, offered by cloud providers and new prices.

The major issue of all cloud services, the contention, when some computational power may be assigned to other processes by the virtualizer, has been studied and presented in articles [54], [56] and [95]. The authors describe the term "flexible CPU", and discuss the function of virtualizers in cloud, which are responsible for assignment of cloud resources to virtual machines (virtual instances). The paper "Resource-freeing attacks: improve your cloud performance (at your neighbor's expense)" [88] presents some interesting techniques that could possibly force the virtualizer to assign more CPU power or RAM to a certain virtual machine (where the sequence of actions, described by the author, has been performed).

Another serious issue, that happens often on most clouds, is the unpredictability of the actual performance of a virtual machine, that is being leased. Articles [58], [82], [83], [85], [89] talk about the performance heterogeneity in cloud services, that is the consequence of the virtualizers' performance assignment. Based on the results, presented in these papers, it can be concluded that every year the cloud performance stability increases and the degree of performance heterogeneity gets lower. The authors of the "Accurate Resource Prediction for Hybrid IaaS Clouds Using Workload-Tailored Elastic Compute Units" [68] paper presents an approach that allows accurate performance prediction of cloud-based virtual machines. Another related paper, [59], discusses the reliability of experiments, performed on virtualized hardware.

The article "CEEM: A Practical Methodology for Cloud Services Evaluation" [7] by Z.Li et al. presents a ten-step methodology for evaluating IaaS cloud services. The authors stated that there is not enough guidelines for cloud services performance evaluation. The authors proposed a generic Cloud Evaluation Experiment Methodology (CEEM). The ten steps of this methodology are briefly described in the following table.

| | Step | Description |
|---|---|---|
| **1** | Requirement Recognition | Recognize the problem, and state the purpose of a proposed evaluation |
| **2** | Service Feature Identification | Identify cloud services and their features to be evaluated |
| **3** | Metrics and Benchmarks Listing | List all the metrics and benchmarks that may be used for the proposed evaluation |
| **4** | Metrics and Benchmarks Selection | Select suitable metrics and benchmarks for the proposed evaluation |
| **5** | Experimental Factors Listing | List all the factors that may be involved in the evaluation experiments |
| **6** | Experimental Factors Selection | Select limited factors to study, and also choose levels/ranges of these factors |
| **7** | Experimental Design | Design experiments based on the above work. Pilot experiments may also be done in advance to facilitate the experimental design |
| **8** | Experimental Implementation | Prepare experimental environment and perform the designed experiment |
| **9** | Experimental Analysis | Statistically analyze and interpret the experimental results |
| **10** | Conclusion and Reporting | Draw conclusions and report the overall evaluation procedure and results |

**Table 3.1**: The Cloud Evaluation Experiment Methodology (CEEM) for cloud services evaluation, as described in the article "CEEM: A Practical Methodology for Cloud Services Evaluation" by Z.Li et al.

The study performed by authors with Google cloud services indicated that CEEM is able to help researches achieve more accurate experimental results.

Apart from the traditional benchmarking procedure, when only one component is being evaluated at a time, there are frameworks that can evaluate the whole system's performance either by running complex tests, or by asking the users, which exactly modules he wants to be tested. Such frameworks have been presented in papers [71], [91], [94]. The article [91] describes the framework CARE (Cloud Architecture Runtime Evaluation), which is intended for cloud platforms evaluation for hosting different types of applications and databases.

The cloud benchmarking for the purpose of deployment of OLTP-based applications requires evaluation of all components of a virtual machine separately as well as the evaluation of their interoperability. This process often implies deployment of some testing application, that can measure the performance of cloud-based virtual machines. This complex procedure has been discussed in articles [55], [65], [70], [79].

"Benchmarking Amazon EC2 for high-performance scientific computing" [45] is probably one of the first papers about Amazon EC2 benchmarking, indicating that there is a significant performance gap between virtual instances and real physical machines. E.Walker, the author of this article, ran several micro- and macro benchmarks to evaluate the performance of high-CPU Amazon instances built in clusters. He used NAS Parallel Benchmark[1], which is a set of tools designed to evaluate performance of computer clusters and MPPTEST benchmark[2] to evaluate bandwidth. Based on the experimental data that he got, he stated that current cloud services were not mature. The author indicated performance gaps between Amazon EC2 instance and local machines in all main components: CPU speed, memory performance and bandwidth. There is a long list of articles [48], [53], [57], [61], [62], [63], [69], [78], [81], that all study the performance of Amazon EC2 virtual machines from the perspective of high performance computing (HPC). These authors perform cloud benchmarking in order to find out, how well are different instances suited for this task. Surprisingly, there are not so many articles about cloud benchmarking for HPC of other cloud providers. [84] presents the results of similar study for Google cloud, and [87] - for Windows Azure.

The article "WPress: An Application-Driven Performance Benchmark For Cloud Virtual Machines" [35] by A. Borhani et al. provides an example of how an OLTP application can be used for performance evaluation of a virtual instance. The author used WordPress, a popular blogging software to evaluate network performance and CPU speed of virtual machines in Amazon EC2, Windows Azure and Rackspace. WordPress was one part of a benchmarking system. The second part was a load generator, installed on a separate machine. The values, measured by WordPress benchmark system, were response time and CPU load. Combined with the cost of a particular instance, this data helped to evaluate the cost-performance ratio of IaaS cloud providers.

There are articles, which present results of the network performance benchmarking. For example, "Network capabilities of cloud services for a real time scientific application" [72], tests how well virtual machines in cloud are suited for hosting real-time applications. Two more general articles, "Choreo: network-aware task placement for cloud applications" [73], and "Empirical evaluation of latency-sensitive application performance in the cloud" [50], evaluate the network performance and measure such values as network latency and bandwidth.

A bit less studied component of cloud-based virtual machines is storage. There are two articles that present a comprehensive research of the storage performance in cloud clouds. The paper "Benchmarking the Storage Services of the Azure Cloud Platform" [47] presents the storage performance in the context of high performance computing. Unfortunately this paper doesn't present a storage performance comparison with other cloud providers. Another article [46], that also deals with the cloud storage, presents an automated approach for cloud storage selection,

[1] https://www.nas.nasa.gov/publications/npb.html
[2] http://www.mcs.anl.gov/research/projects/mpi/mpptest/

based on the performance of that cloud.

## 3.3 Contributions Of This Work

The related work provided a background for conducting this research. This work uses several tools and a benchmarking framework, which had been marked by other researches as highly accurate and well suited for benchmarking virtual machines in general and cloud-based VMs in particular. The outcome of this work is the actual performance data of current Amazon EC2 and Google GCE virtual machines. But the actual performance and costs analysis are prone to fast changes due to hardware upgrade in providers' data centers and prices getting lower, and therefore this research also includes the analysis of the data gathered by the benchmarking tools. This analysis explains why certain virtual machine type has different performance compared to the similar VM type, offered by another provider. Research, performed prior to this work, helped to understand the technology behind the cloud-based virtual machines, the use and functions of virtualizers in data centers and techniques for hardware assignment. The analysis, presented in this work, can be applied to different cloud providers and different types of cloud-based virtual machines, and this makes this work valuable for future researches in this field.

# Chapter 4

# Study Setup

This chapter discusses the central part of this work - setting up and configuring different benchmarking tools for cloud performance evaluation. Benchmarking software has to be installed, configured and adopted for use in Amazon and Google cloud infrastructure. All experiments had been run on virtual machines of different types, leased from these two cloud providers. They were accessed via SSH or terminal emulator in a web browser, and all configuration tasks were performed directly on these VMs.

Most benchmarks used in this work require two virtual machines to be used. RUBiS requires a client and a server virtual instances. JMeter requires a target machine and a load generating machine. Iperf also requires two machines for measuring the bandwidth. For the sake of results reliability, two virtual machines of one benchmark had been created in the same cloud. The reason for that is because in this case two machines are in the same LAN and have stable and high-speed link between them.

The sections in this chapter provide detailed information about each benchmark used in this work. The chapter has following structure. First, the "top", that is a built-in system monitoring tool and its application in this work, is described. Then, micro benchmarks, which evaluate the performance of a separate component in isolation, are discussed. They include benchmarking tools from Phoronix Test Suite and Iperf. After that, the combination of JMeter benchmark and a WordPress blogging software is covered. Finally, the most complex benchmark used in this work - framework RUBiS, is presented in the last section of this chapter.

## 4.1 "Top" Command in Linux

This section covers the use of Linux "top" command in context of the this work.

"Top" command represents the system summary information such as the percentage of CPU being used for certain tasks, idle CPU, RAM state and a list of tasks of different users. The most important data from the perspective of this thesis is represented in the line that shows CPU load. In particular, fields "id" and "st" are of the most importance for this work.

**"id",** or "idle", is the percentage of the CPU not used. Based on this number, it is possible to calculate what percentage of CPU is being used at the moment;

**"st",** or "steal time", is the amount of CPU, "stolen" from this virtual machine by a hypervisor for other tasks (such as running another virtual machine). It will be 0 on desktop or a physical server without virtual machines running.

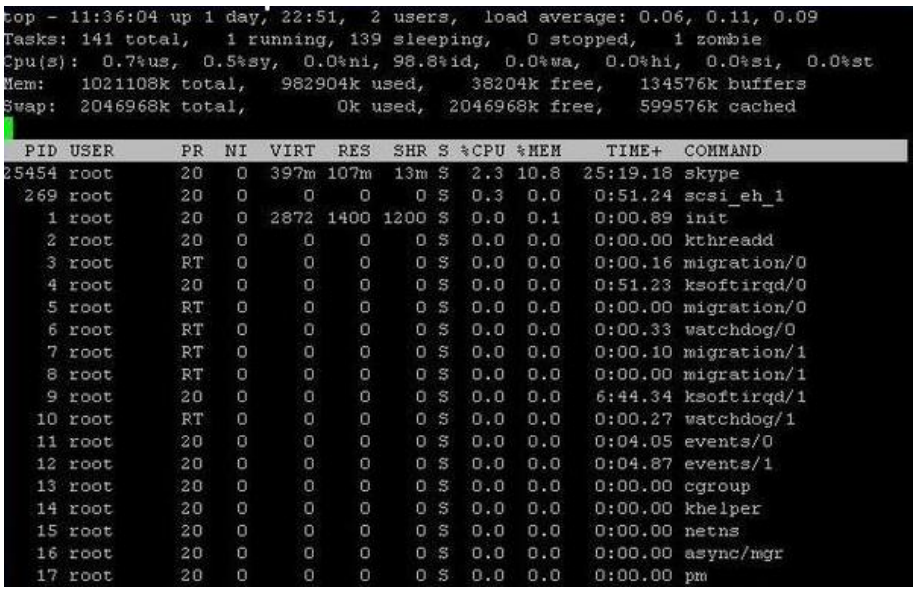


**Figure 4.1**: Example of "top" command in Linux

The "top" command was used throughout the work in combination with other benchmarks to estimate the CPU load and the percentage of "stolen" CPU.

## 4.2 Phoronix Test Suite

Phoronix Test Suite[1] is a collection of micro benchmarking tools developed for Linux operating systems. Phoronix comprises a large number of benchmarks for different purposes.

### 4.2.1 Overview

The Phoronix Test Suite is a software platform that comprises a large number of micro benchmarks aimed to evaluate performance of separate hardware components. Phoronix framework and included tools are constantly updated, and new tools can be easily added. The platform is designed to be used by a large number of users, and therefore the installation and configuration process is straightforward, and also the usage guidelines are provided.

The Phoronix Test Suite is based on a large number of small tools used for hardware components benchmarking. Phoronix project was created in 2004 as a universal testing framework. Phoronix software is an open source product.

During first few years Phoronix supported only Linux based systems, but later support for Apple OS X, Microsoft Windows, BSD, and Solaris OS was added. All new tools that had been added, were accompanied with detailed user guides and manuals that make the benchmarking process for end users easier.

[1] http://www.phoronix-test-suite.com/

### 4.2.2 Selection of Benchmarks

Phoronix Test Suite comprises a large number of benchmarks, intended to evaluate performance of different components of computer hardware.

#### CPU Performance

The following benchmarks test the CPU performance of a virtual machine. These particular benchmarks are the most representative for CPU testing, and they are popular among researchers and developers dealing with hardware performance evaluation [39], [40], [67]. Also, these benchmarks eliminate any use of RAM for pure CPU evaluation.

**C-Ray**
The C-Ray[2] benchmark is a ray tracer that tests the CPU performance based on floating-point calculations. C-Ray is capable of testing performance of multiple threads. The benchmark generates an image of a 1600x1200 resolution.

C-Ray generates very small amount of data, ensuring that RAM doesn't participate in the process of generating images in any way. This allows to perform an isolated CPU performance test [38].

**Gzip**
Gzip[3] is a file compression tool. This benchmark measures the time taken to compress a sample file. Advantages of Gzip are high performance (the need to use any amount of RAM is eliminated). The size of the sample file is small to ensure that RAM and other hardware components don't affect CPU performance test.

#### RAM Performance

The following two benchmarks were selected for RAM performance evaluation.

**Ramspeed**
Ramspeed is a benchmarking tool that tests the speed of RAM, returning a value represented in megabytes per second. The memory speed is calculated based on memory frequency in MHz and RAM timings. The higher the number, the better is the performance [38]. Ramspeed benchmark gives more precise results compared to many other RAM benchmarking tools, it is also more compact [41].

**Bork**
Bork[4] evaluates both RAM and CPU performance combined. Bork is a small, cross-platform file encryption utility. The test measures the amount of time it takes to encrypt a sample file. Bork keeps the files it encrypts in RAM, and that is the reason why RAM affects the benchmarking results. The tool has minimal external dependencies and doesn't rely on other hardware components.

This selection of benchmarks from Phoronix Test Suite is based on the requirement that only a single component (or a combination of CPU and RAM in case of Bork) should be tested. Therefore complex benchmarking platforms were not considered in this part of the work. These simple

[2] https://openbenchmarking.org/test/pts/c-ray
[3] https://openbenchmarking.org/test/pts/compress-gzip
[4] http://openbenchmarking.org/test/pts/bork

benchmarks, which are well established and proved their reliability and stability, allow performing pure CPU and RAM performance evaluation of virtual machines.

## 4.3 Iperf Benchmark

Iperf[5] is a network testing tool that evaluates the bandwidth of a network. Iperf can generate TCP and UDP packet streams depending on its settings.

### 4.3.1 Overview

Iperf is a C-based networking tool. Users can set different parameters depending on the experiment requirements and network type. For evaluating the network throughput, Iperf has to be installed on at least two systems, one of which is a server, hosting Iperf instance and waiting for a client machine to connect to it to test the bandwidth. Iperf is an open source tool, and can be used on Unix and Windows based systems.

**UPD.** In UDP mode, a user can specify the datagram size and measure datagram throughput.
**TCP.** In TCP mode, a user can measure the payload throughput.

The network parameters that can be evaluated by Iperf are bandwidth, latency, jitter and datagram loss.

### 4.3.2 Installation and Configuration

The installation process is relatively simple. All that required is to download the installation package from official website using "wget" command, make this file executable and move it to the correct directory. Since Iperf has server-client functionality, the installation and configuration procedures should be performed on both machines.

After that, Iperf commands with different parameters depending on the required settings mode should be executed to initiate the Iperf session. The commands should be executed on client machine. Server always stays in listening state.

## 4.4 JMeter Benchmark

Apache JMeter[6] is a load generator and a client simulator software.

### 4.4.1 Overview

Apache JMeter™ is an open source software intended for load testing of web servers and web applications, and for measuring their performance.

JMeter can be used to test the performance of web services, web languages (PHP, ASP.NET), databases, FTP servers, etc. JMeter can simulate different types of load on a server. It allows simulating any amount of virtual clients that can perform different behavior (browsing to different pages, accessing files). Results can be presented in graphical, tabular or textual form [32].

[5] http://iperf.fr/
[6] http://jmeter.apache.org

JMeter is owned by Apache since 2011. Before that it was a small scale project supported by enthusiasts.

### 4.4.2 Installation

Installation of JMeter is relatively simple. All that required is to download the installation files on a Linux machine and install it following a standard installation procedure. In case of Windows OS, the JMeter files have to be downloaded, and the program can be started immediately. Installation should be done on a client machine, which will generate the load on server and simulate clients [34]. The server machine doesn't have to have a JMeter installation, or any other special software. It just needs to run a web service that will be tested.

### 4.4.3 Test Plan Configuration

The most interesting part in JMeter installation and configuration is creation of a test plan. A researcher can set all parameters according to his needs and requirements of the experiment. There are two ways how a test plan can be created: using a command line with specific JMeter configuration commands, or using a GUI provided with the JMeter installation package. The creation of a test plan and reasons for choosing certain parameters are discussed in more detail later in this section.

Since the virtual machines leased from Amazon and Google had a server installation of Linux and the only way of working with them was using a command line interface, the test plane should have been created in it. But it turns out that the file that contains a test plan configuration, has the same structure if created using command line interface or using GUI. Therefore, a windows machine was used to create a test plan of the experiment. There exist a number of research papers and articles published that cover the functionality and configuration of JMeter [33], and modern guidelines published by Apache[7] help to understand the way how a test plan should be configured, and the functionality of a GUI of JMeter.

The test plan created for this experiment was configured to access the website by its URL(in this case a WordPress blog, discussed later) five times in a row. This is because of the connection establishment procedures which take place when requesting a new URL for the first time (ARP process, mac address caching). During the first access attempt the page load time and latency values are high. At the second attempt these values get lower, but they are still not stable. And starting from the third attempt both page load time and latency values become stable. JMeter doesn't cache web pages to provide real page load time in all attempts.

### 4.4.4 WordPress Overview

In order to evaluate performance of the server, the server has to run some web application or to have a website, that will be accessed by a JMeter installed on a remote client machine. WordPress[8] was selected to be such a web application in this work.

WordPress as a widely used open-source blogging software [35], [70, [79]. It is often used in combination with a benchmark to measure server performance on a real data (a blog or a website) [36]. In this work the WordPress application was configured as a web blog with 232 posts, 1272 comments and 100 registered users.

[7] http://www.apache.org/
[8] http://www.wordpress.com

### 4.4.5 Wordpress Installation and Configuration

The process of installation of WordPress is straightforward, although it requires some knowledge of Linux operating systems, and it is described in many IT related articles [37]. The process includes downloading the WordPress installation package, running installation script, configuring MySQL database to work with WordPress on virtual computer, configuring PHP to work with WordPress, creating a dedicated user and modifying his rights. The final step is a web based configuration of a blog name and settings, which can be done remotely in a web browser.

After the basic installation and configuration, the Demo Data Creator[9] plugin was installed in the WordPress environment. It can generate any amount of content (users, categories, posts, comments, web pages) in the blog. The posts in this case are random articles, some of them include pictures, polls, and other interactive elements.

Once the initial configuration of the blog has been performed, the snapshot of the virtual machine has been taken to reproduce it on other virtual machines of different types. In this way, the content of a blog remains the same for all experiments within this work.

## 4.5 RUBiS Benchmark

RUBiS is a simulated auction website, very similar to eBay.com. It is used to evaluate application server performance.

### 4.5.1 Overview

The RUBiS benchmark first appeared in 2002 with version. There were multiple releases in 2002 - 2004, but after 2004 all further development stopped. The latest version is 1.4.2.

RUBiS benchmark simulates an auction website: placing items for sale, bidding, rating, commenting and selling items. There are three types of users - visitor, buyer and seller. Visitors can only browse the items that are placed. Buyers can register, browse items, leave comments, bid for items and view comments and ratings left by other users. Sellers, apart of that, can place or delete items and specify price for an item.

RUBiS platform, when installed, can be opened in a web browser for testing purposes. Fur automating the benchmarking process, the second machine, client, has to be set up and configured. Client emulates user behavior for different user workload patterns, and it displays the server performance data. The figure below represents RUBiS functionality. Virtual clients generate load on web servers, which load application server, which in turn query data from the database.

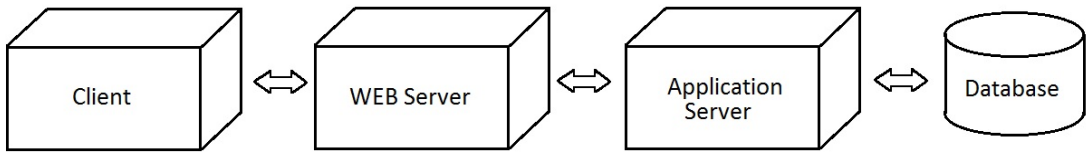


**Figure 4.2**: RUBiS benchmark overview

[9] http://wordpress.org/plugins/demo-data-creator/

The auction site has a number of actions that can be performed from the client's web browser (or from a remote client emulator machine). Among the most important ones are browsing items by category or region, bidding, buying or selling items, leaving comments and others. Browsing items includes opening a web page, where the item's description is located, closing it and moving on to the next item. At the same time, all current bids for these items are displayed. Bidding includes same actions as for browsing, plus putting a bid and competing with other buyers for the item. The combination of these two behavior types represents the real auction site behavior, where some users bid for items, and others only browse them.

The interaction between clients and a server is done by establishing a session for each simulated user. For establishing a session, HTTP connection is required. User sessions last different time, emulating real user behavior, when users spend different time on reading description and taking a decision.

RUBiS uses MySQL database for storing its data. There are 7 tables in the database: users, items, categories, regions, bids, buy_now and comment. Users have following attributes: name, nickname, password, region, rating and balance. Bids have following attributes: seller, bid, and a max_bid value. Items that are bought via buy-it-now option, are stored in the buy_now table. The comments table stores comments about items.

The RUBiS benchmark is the largest and the most complex benchmark in terms of installation, configuration and execution processes, used in this work, and it was intended to evaluate the overall performance of virtual machines from Amazon and Google. RUBiS doesn't measure CPU or RAM performance explicitly, but it provides an auction website performance results in form of requests per second. A request is a single action of a simulated user. It can be browsing, bidding, leaving a comment, buying or any other actioned implemented in the framework.

At the same time, the CPU load was measured while RUBiS benchmark was performing. it was measured with the help of the system monitoring tool "top", implemented in Linux operating systems.

### 4.5.2 Installation And Configuration

The process of installation, configuration and running RUBiS benchmark is challenging from many perspectives. First, it is not just "download and install" procedure. Researcher has first to set up a LAMP environment (Linux, Apache, MySQL and PHP) with Java to install the RUBiS benchmark on. Then, there is a number of steps that he has to do in order to adopt the downloaded RUBiS files to the local machine, like editing configuration files in order to specify database user name and password and all necessary system paths. After that, RUBiS data generation scripts have to be run in order to generate tables with regions, items, users, categories and so on. Each of these scripts has to be adopted to a specific environment, i.e. some variables have to be changed.

The process of installation and configuration itself is not complicated, but there are no official guides or instructions that document what has to be done in order to set up the benchmark. The steps specified in previous paragraph were discovered at a forum, where one RUBiS enthusiast shared his experience of working with RUBiS benchmark [30]. But there were multiple misconfigurations and missed errors, which were later noted by Dr. Philipp Leitner in his RUBiS installation tutorial [31]. Apart of these two sources there are no comprehensive guides for RUBiS benchmark installation and configuration.

Another problem, which has been encountered during the RUBiS configuration, was the required number of simulated virtual users. This number is the parameter that sets the load on the server benchmark. Therefore it is important to find the optimal value, which loads the CPU of the virtual machine of the smallest type close to 100 per cent. In this case it is possible to see the changes of the CPU load and changing performance test data from one instance type to another.

Since one RUBiS benchmark run takes about 20 minutes, the process of determining the optimal user number takes a long time. The optimal number of users in benchmark was found to be in range between 5000 and 8000. Experiments in this work were conducted for both values.

The RUBiS benchmark comes with most of its parameters preconfigured to their default values. There are already 62 regions created (Mostly U.S. states, and some other territories), 20 categories of items similar to eBay, sample item description text and few others. These default values haven't been changed in this work, because they already satisfy the basic requirements to the auction platform. The number of items, bids and comments depends on the number of simulated users, specified by researcher. These parameters are generated by the script after all other non-user dependent parameters have been specified.

| Parameter | Value |
|---|---|
| Number of Users | 5000 or 8000 |
| Regions | 62 |
| Categories | 20 |
| Number of Items | 33000 |
| Item Description Size, byte | 8192 |
| Number of Bids | 330000 |
| Number of Comments | 500000 |

**Table 4.1**: RUBiS parameter values used in this work

RUBiS database used in this work contains 33000 items for sale, distributed among 20 categories and 62 regions. A history of 500000 auctions is kept in the old-items table. There is an average of 10 bids per item, or 330000 entries in the bids table. The buy_now table is relatively small, because less than 10 per cent of the items are sold without auction ("Buy-it-now"). Users leave comments for 95 per cent of their transactions. The comments table contains about 500000 comments to current items and old items. The total size of the database is around 1.4 GB. All parameter values, both default ones and selected after some research and analysis, are presented in the table.

The result of the RUBiS benchmark is the "requests per second" characteristic, which represents how many requests the client machine can remotely initiate to the server machine.

After the initial installation of RUBiS benchmark it turned out that one of the internal timers in the client emulator script was set to some ambiguous large number instead of 120 seconds, which it should have been set to according to the documentation, which made the execution process impossible. This case, an error in variable mentioned above, made the process of troubleshooting and making things work complicated and time consuming, because it required the analysis of the source code of the whole RUBiS benchmarking framework.

It worth mentioning that the whole benchmarking software is written in Java, and the quality of the source code structure doesn't allow easy understanding of its functions and procedures. As mentioned before, there were also other errors in variables' names, which were pointed out in tutorials [30], [31].

## 4.6 Methods of Data Collection in This Work

All experiments within this work had been run a certain number of times depending on a benchmark to guarantee data accuracy and independence from such factors as different performance at different time of the day, connection disruption, possible dysfunction of a virtual machine,

network connection, etc.

Close attention was paid to the contention described in the Background chapter. To avoid this effect, each virtual machine was recreated three times, and same experiments were performed on it to get the average value. Later, results of their performance benchmarking were compared and analysed to get accurate results.

The final value of a certain factor was calculated using the following method. First, the experiments had been run three times (five times in case of JMeter and Iperf) in the same conditions. The average value was calculated at this stage. Then, at another time, new exactly the same virtual machine was created, and the same experiments had been run the same number of times on it. Second average value had also been calculated for this machine. Later, third same virtual machine was created, and same experiments had been run again, and the average value was calculated. Finally, the Grand Average value had been calculated, based on the three average values from all experiment sessions described above. This Grand Average has been taken as the final value of a certain performance factor.

At the same time, all intermediate results, such as average values for a particular virtual machine, were analysed separately and compared with the results of other virtual machines of the same type, but created at different time.

## 4.7 Overview of Cloud-Based VMs, Used in This Work

The two tables below give an overview of virtual machines from Amazon EC2 and Google GCE clouds, used in this work.

| | CPU Type | CPU Speed, GHz | RAM, GB | Disk, GB |
|---|---|---|---|---|
| **Small** | Intel Xeon | 2.0 (1 core) | 2 | 10 |
| **Medium** | Intel Xeon | 2.0 (1 core) | 4 | 10 |
| **Large** | Intel Xeon | 2.0 (2 core) | 8 | 10 |
| **Xlarge** | Intel Xeon | 1.8 (4 core) | 16 | 10 |

**Table 4.2**: Amazon EC2 virtual machines used in this work

| | CPU Type | CPU Speed, GHz | RAM, MB | Disk, GB |
|---|---|---|---|---|
| **Small** | Intel Xeon | 2.5 (1 core) | 1740 | 11 |
| **Medium** | Intel Xeon | 2.5 (1 core) | 3840 | 11 |
| **Large** | Intel Xeon | 2.5 (2 core) | 7680 | 11 |
| **Xlarge** | Intel Xeon | 2.5 (4 core) | 15360 | 11 |

**Table 4.3**: Google GCE virtual machines used in this work

All these VMs are categorized as "general purpose" virtual instances in both Amazon and Google. They are optimized by all parameters and are intended to be used for a wide variety of tasks.

# Chapter 5

# Results and Data Analysis

This chapter presents the results produced by all benchmarks deployed within this work, and provides some discussion based on the outcome of the experiments. The chapter is built in the following way. Each section presents results and analysis of a certain benchmark used in this work. Also each section discusses results of Amazon and Google virtual machines and compares them. This chapter has similar structure to the previous one. First, results of micro benchmarks are presented. They include Phoronix Test Suite and Iperf. Then follow the results of performance evaluation by JMeter with WordPress. Finally, the results of RUBiS benchmark are discussed at the end. In the last section, the correspondence of the performance data of cloud-based virtual machines and physical machines is presented to state whether they have similar performance or not.

## 5.1 Phoronix Test Suite Benchmarking

As mentioned in previous chapter, four benchmarks from the Phoronix Test Suite were used in this work to evaluate different components of virtual instances from Amazon and Google. This section consists of four parts, each discussing the results of these four benchmarks.

### Gzip Benchmark

Gzip benchmark evaluates the CPU performance and its output is the time needed to compress a sample time. The two tables below present the results of running Gzip benchmark on Amazon and Google virtual instance.

| | **Run 1, sec** | **Run 2, sec** | **Run 3, sec** | **Average, sec** |
|---|---|---|---|---|
| **Small** | 55.9 | 56.1 | 54.6 | **55.5** |
| **Medium** | 25.6 | 25.2 | 25.1 | **25.3** |
| **Large** | 25.3 | 25.1 | 25.2 | **25.2** |
| **Xlarge** | 27.4 | 26.9 | 26.8 | **27.0** |

**Table 5.1**: Gzip benchmarking results for Amazon EC2 instances

| | Run 1, sec | Run 2, sec | Run 3, sec | Average, sec |
|---|---|---|---|---|
| **Small** | 54.6 | 52.5 | 56.8 | **54.6** |
| **Medium** | 16.7 | 16.7 | 16.6 | **16.7** |
| **Large** | 16.4 | 16.4 | 16.3 | **16.4** |
| **Xlarge** | 16.1 | 16.0 | 16.0 | **16.0** |

**Table 5.2**: Gzip benchmarking results for Google GCE instances

Based on the experimental results, we can conclude that instances of Small type from both Amazon and Google have same low performance compared to larger instances. Such a sharp difference in performance can be explained by the fact that according to the "top" monitoring tool in Linux, the amount of "stolen" CPU by the virtualizer was about 45 per cent for Small instances, while for larger instances this value was always less than 1 per cent. In this case almost a half of CPU power was reassigned by the virtualizer to some other processes. Another fact that draws attention is that Medium, Large and Xlarge instance types have similar performance data. This is the consequence of the Gzip benchmark, which runs only on one core. This is also the reason of the difference in performance between Amazon and Google instances - Amazon uses CPUs with 2.0 GHz, but Google uses CPUs with 2.5 GHz speed.

The figure below provides visual representation of performance differences between Amazon EC2 and Google GCE instances. For its Xlarge instance Amazon uses Intel Xeon processor with 4 cores by 1.8 GHz each. This is noticeable on the graph - the Gzip compressing time is higher for Xlarge Amazon instance compared to Medium and Large types (only one core is used).

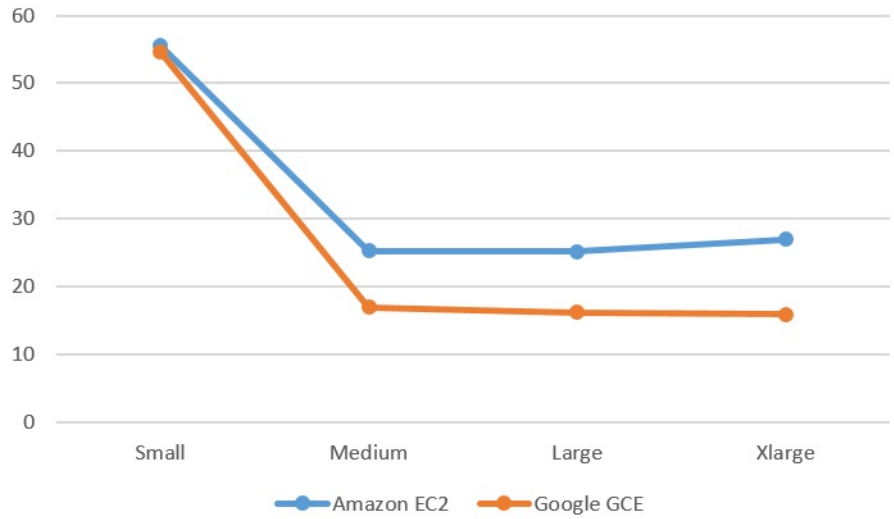


**Figure 5.1**: Gzip benchmarking results for Amazon EC2 and Google GCE instances, in seconds

## C-Ray Benchmark

C-Ray benchmark is intended for CPU performance evaluation by the use of floating point calculations. The tables below present benchmarking data for Amazon EC2 and Google GCE instances.

| | Run 1, sec | Run 2, sec | Run 3, sec | Average, sec |
|---|---|---|---|---|
| **Small** | 361.4 | 372.1 | 367.5 | **367.0** |
| **Medium** | 272.6 | 275.6 | 268.2 | **272.1** |
| **Large** | 214.9 | 209.7 | 210.0 | **211.5** |
| **Xlarge** | 143.7 | 145.6 | 148.8 | **146.0** |

**Table 5.3**: C-Ray benchmarking results for Amazon EC2 instances

| | Run 1, sec | Run 2, sec | Run 3, sec | Average, sec |
|---|---|---|---|---|
| **Small** | 353.9 | 357.2 | 355.0 | **355.4** |
| **Medium** | 178.9 | 178.8 | 178.4 | **178.7** |
| **Large** | 135.9 | 135.8 | 135.8 | **135.8** |
| **Xlarge** | 67.9 | 67.9 | 67.9 | **67.9** |

**Table 5.4**: C-Ray benchmarking results for Google GCE instances

General trend is similar to the Gzip benchmark, apart of the fact the C-Ray is capable of running on all cores. The performance of a Small instance is relatively low due to the fact that about 45 per cent of its power is "stolen" by the virtualizer according to the "top" system monitoring tool. The amount of "stolen" processing power is around 5 per cent lower at Google instances compared to Amazon. comparison of Medium, Large and Xlarge instance types from Amazon and Google has same patterns as demonstrated earlier with the Gzip benchmark. Due to the CPU speed difference (2.0 GHZ at Amazon and 2.5 GHz at Google), Google instances have better performance. All these trends are demonstrated on the figure on the next page.

The table after the graph presents the price-performance ratio for virtual machines in Amazon and Google clouds, represented in computational values. Higher value means better ratio. This data is calculated in following way. Since the smaller time of the benchmark implies better performance, then the inverse of the time for each VM was divided by the corresponding machine price. As can be seen from the table, virtual machines in Google cloud have better price-performance ratio.

| | Amazon EC2 | Google GCE |
|---|---|---|
| **Small** | 0.036 | 0.067 |
| **Medium** | 0.024 | 0.066 |
| **Large** | 0.015 | 0.044 |
| **Xlarge** | 0.010 | 0.042 |

**Table 5.5**: Price-performance ratio for different VM types from Amazon EC2 and Google GCE based on C-Ray benchmarking, in computational values

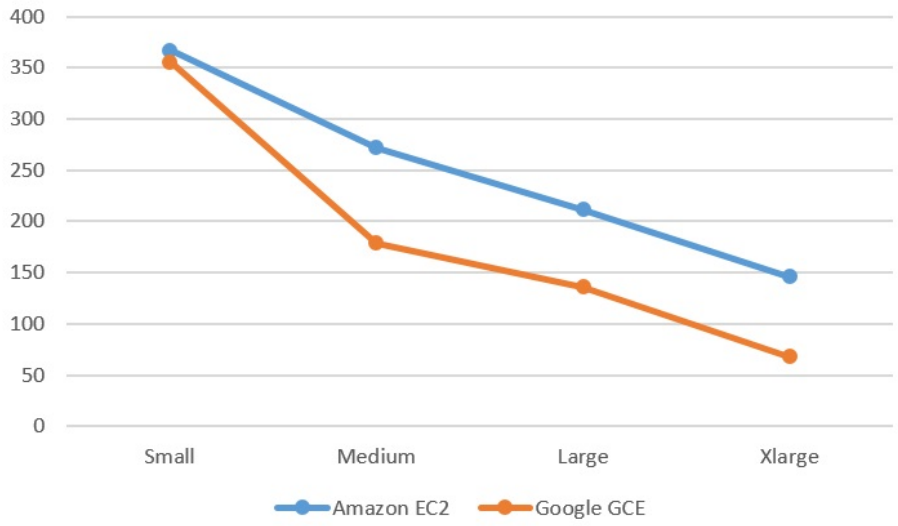


**Figure 5.2**: C-Ray benchmarking results for Amazon EC2 and Google GCE instances, in seconds

## Bork Benchmark

Bork is an encryption utility and the benchmark evaluates both CPU and RAM performance. The tables below represent benchmarking data for Amazon EC2 and Google GCE instances.

| | Run 1, sec | Run 2, sec | Run 3, sec | Average, sec |
|---|---|---|---|---|
| **Small** | 71.2 | 71.4 | 71.7 | **71.5** |
| **Medium** | 61.3 | 61.4 | 61.3 | **61.3** |
| **Large** | 28.8 | 28.7 | 28.4 | **28.6** |
| **Xlarge** | 28.4 | 28.2 | 28.0 | **28.0** |

**Table 5.6**: Bork benchmarking results for Amazon EC2 instances

| | Run 1, sec | Run 2, sec | Run 3, sec | Average, sec |
|---|---|---|---|---|
| **Small** | 118.3 | 121.8 | 119.3 | **119.8** |
| **Medium** | 50.8 | 56.9 | 53.0 | **53.6** |
| **Large** | 28.4 | 27.1 | 27.8 | **27.8** |
| **Xlarge** | 21.8 | 19.1 | 19.9 | **20.2** |

**Table 5.7**: Bork benchmarking results for Google GCE instances

Based on the data from the tables, it is not easy to determine a pattern. The performance of the instance of type Small is better at Amazon (encryption time is less), but larger instances from Google perform slightly better than from its competitor. Such a difference can be explained by the amount of memory that instances have. If Small instance at Amazon has exactly 2 gigabytes of RAM, at Google it has only 1740 megabytes. At Amazon's computing center the virtualizer in

a best case will assign a single 2GB RAM chip to one Small instance without the need to accommodate another instances on the same chip. At Google's computing center the virtualizer will most likely combine these 1740 megabytes from multiple chips (hardly ever the instance will get all 1740 megabytes on one chip due to preservation policies). In this case the RAM performance obviously decreases.

But with larger instances, which have more than 4 gigabytes of RAM, the effect of combining memory from multiple chips decreases, since the biggest amount of RAM will reside on a single chip. For example, in case of Google's Xlarge instance that has 15360 megabytes of RAM, 12 Gb will most likely not be located on chips that are shared with other processes. At Amazon, that assigns only even number of gigabytes of RAM, pure RAM performance is better (this fact is confirmed by the Ramspeed benchmark, presented later), but since the Bork benchmark evaluates the performance of both CPU and RAM, better CPU of Google's VMs compensate relatively poor performance of RAM. The figure helps in understanding of the non-trivial dependencies of CPU and RAM performance of virtual machines in Amazon EC2 and Google GCE.

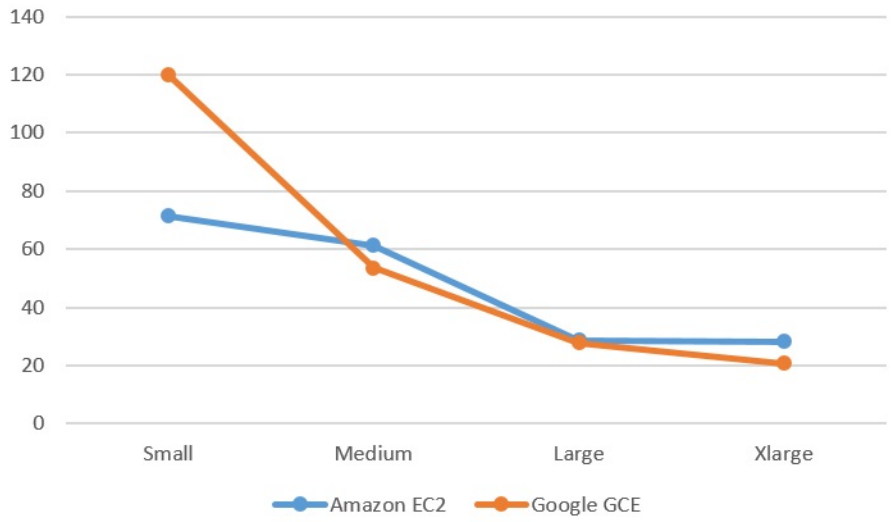


**Figure 5.3**: Bork benchmarking results for Amazon EC2 and Google GCE instances, in seconds

## Ramspeed Benchmark

Ramspeed benchmark tests RAM performance, and its output is the speed in megabytes per second. The two tables below represent the results of benchmarking of virtual machines in Amazon EC2 and Google GCE using Ramspeed benchmark.

Since Ramspeed is a pure RAM benchmark, the difference between Amazon and Google virtual instances is more distinguishable here, especially on larger instances. Due to the fact that Amazon instances have only even number of RAM gigabytes, and in most cases separate hardware chips are assigned to them, virtualizer does not have to combine memory from multiple chips, unlike with virtual instances from Google.

| | Run 1, MB/s | Run 2, MB/s | Run 3, MB/s | Average, MB/s |
|---|---|---|---|---|
| **Small** | 2914.84 | 3050.18 | 2968.02 | **2977.68** |
| **Medium** | 8173.37 | 8215.11 | 8214.40 | **8200.96** |
| **Large** | 15307.80 | 16168.47 | 16185.79 | **15887.35** |
| **Xlarge** | 15762.77 | 15959.05 | 16144.94 | **15955.59** |

**Table 5.8**: Ramspeed benchmarking results for Amazon EC2 instances

| | Run 1, MB/s | Run 2, MB/s | Run 3, MB/s | Average, MB/s |
|---|---|---|---|---|
| **Small** | 5875.95 | 5860.42 | 4879.19 | **5538.52** |
| **Medium** | 8667.95 | 8554.28 | 8571.39 | **8597.87** |
| **Large** | 9321.51 | 9225.20 | 9124.19 | **9223.63** |
| **Xlarge** | 13881.79 | 13875.53 | 13961.95 | **13906.42** |

**Table 5.9**: Ramspeed benchmarking results for Google GCE instances

However, we can notice that RAM performance of Google's Small instance is better than Amazon's. While doing CPU benchmarking, the amount of "stolen" CPU speed for Amazon instance was much higher than for Google instances. This level was close to 50 per cent. It is recommended not to use Small general purpose instances for tasks that require high stability and preciseness in performance values [3], [8], [44]. It implies that virtualizers at Amazon's computing centers combine multiple pieces of hardware to build a VM of type Small. Obviously, performance of such machines will be lower than performance of a machine that was built on a single piece of hardware.

The figure displays graphs of RAM performance of Amazon and Google instances.

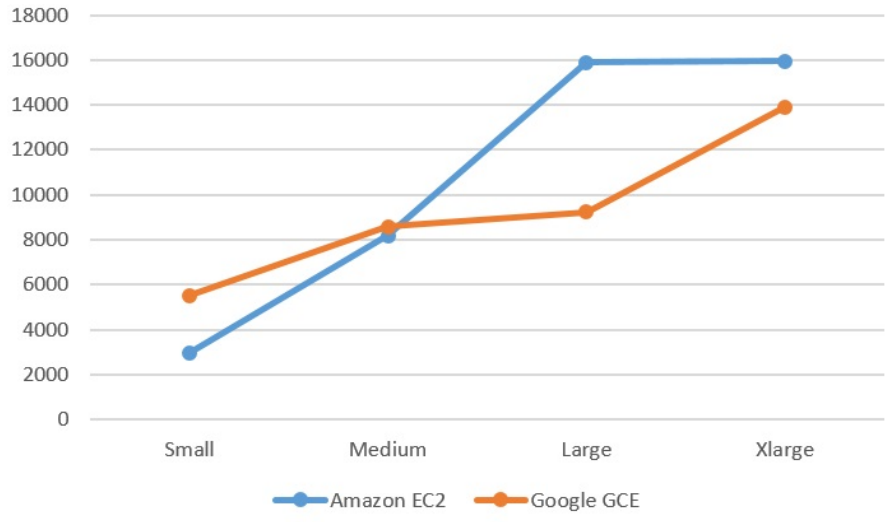


**Figure 5.4**: Ramspeed benchmarking results for Amazon EC2 and Google GCE instances, in megabytes per second

| | Amazon EC2 | Google GCE |
|---|---|---|
| **Small** | 75324 | 66700 |
| **Medium** | 105194 | 202380 |
| **Large** | 207792 | 221420 |
| **Xlarge** | 209460 | 166664 |

**Table 5.10**: Price-performance ratio for different VM types from Amazon EC2 and Google GCE based on Ramspeed benchmarking, in computational values

The table above presents the price-performance ratio for different virtual machine types, used in this work, based on the results of the Ramspeed benchmark. Since the higher speed implies better performance, the speed in MB/s was divided by the VM price and multiplied by the amount of RAM that the machine has. The result is displayed in computational values. These values are not related to the computational values, used in previous section. According to these results, VMs in Amazon and Google clouds have close ratio of the Small and Large virtual machine types.

## 5.2 IPerf Benchmarking

IPerf benchmark was used in this work to measure bandwidth on a link between two virtual machines. The tables below present the data gathered using IPerf benchmark.

| | Run 1, MBit/s | Run 2, MBit/s | Run 3, MBit/s | Average, MBit/s |
|---|---|---|---|---|
| **Small** | 629 | 651 | 664 | **648** |
| **Medium** | 881 | 858 | 859 | **866** |
| **Large** | 898 | 920 | 917 | **912** |
| **Xlarge** | 1073 | 1096 | 1102 | **1090** |

**Table 5.11**: IPerf benchmarking results for Amazon EC2 instances

| | Run 1, MBit/s | Run 2, MBit/s | Run 3, MBit/s | Average, MBit/s |
|---|---|---|---|---|
| **Small** | 1331 | 1381 | 1389 | **1367** |
| **Medium** | 1425 | 1391 | 1390 | **1402** |
| **Large** | 1591 | 1580 | 1594 | **1588** |
| **Xlarge** | 1617 | 1642 | 1633 | **1630** |

**Table 5.12**: IPerf benchmarking results for Google GCE instances

From this data it is clear that VMs of all types in Google cloud have much higher bandwidth. Moreover, the bandwidth does not change significantly from one instance type to another. The

bandwidth of Small virtual machine is only about 20 per cent less than the bandwidth of the Xlarge machine. For Amazon instances, Small VM type has about 70 per cent less bandwidth than the Xlarge machine. The interesting fact here is that Amazon assigns "moderate" networking performance to both Medium and Large types according to its website. Most users assume it to be identical. But in fact, bandwidth of a Medium instance is 46 MBit/s (5.4 per cent) less than bandwidth of a Large instance. It is not a large difference, and it can even be considered as error. But the JMeter benchmark described in the next section, confirmed this finding and discovered even more significant difference in networking performance of these two instance types from Amazon. The graph below compares bandwidth of different instance types from Amazon and Google based on the data presented in two tables earlier.

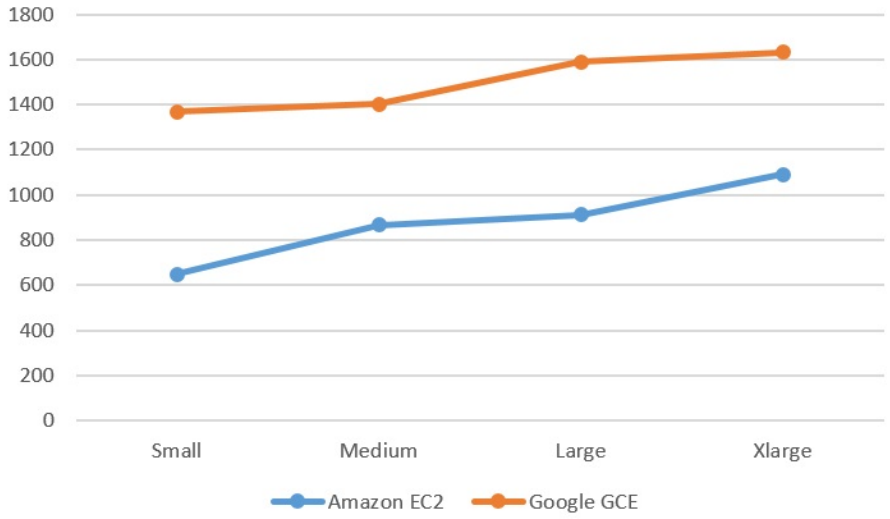


**Figure 5.5**: IPerf benchmarking results for Amazon EC2 and Google GCE instances, in megabits per second

The table on the next page presents the price-performance ratio for different virtual machine types, used in this work, based on the results of the Iperf benchmark. Since the higher bandwidth implies better performance, the bandwidth value was divided by the VM price. The result is displayed in computational values. These values are not related to the computational values, used in previous sections. According to these results, VMs in Google clouds have significantly higher ratio compared to Amazon VMs.

| | Amazon EC2 | Google GCE |
|---|---|---|
| **Small** | 8052 | 32857 |
| **Medium** | 5519 | 16785 |
| **Large** | 2987 | 9553 |
| **Xlarge** | 1769 | 4851 |

**Table 5.13**: Price-performance ratio for different VM types from Amazon EC2 and Google GCE based on Iperf benchmarking, in computational values

## 5.3 JMeter and WordPress Benchmarking

The JMeter benchmark in this work evaluates the network performance of virtual machines of different types in Amazon and Google clouds. As a result of benchmarking, JMeter outputs the page load time and the latency in milliseconds. The target page was the home page of the WordPress blog hosted on the server machine.

### Amazon EC2

The table below represents the results of benchmarking of four different types of Amazon virtual instances. Each "run" in tables comprises the average of last four measurements out of five, performed according to the test plan, described in chapter "Study Setup". The reason why only results of four measurements were considered, is the performance data in the first measurement, which was considerably lower due to networking processes related to accessing a page for the first time (ARP request, mac address caching, etc).

| | | Run 1 | Run 2 | Run 3 | Average |
|---|---|---|---|---|---|
| **Small** | **Load Time, ms** | 2620 | 2460 | 2480 | **2520** |
| | **Latency, ms** | 1085 | 1080 | 1125 | **1097** |
| **Medium** | **Load Time, ms** | 1220 | 1190 | 1270 | **1227** |
| | **Latency, ms** | 623 | 615 | 660 | **632** |
| **Large** | **Load Time, ms** | 950 | 700 | 800 | **817** |
| | **Latency, ms** | 540 | 620 | 420 | **527** |
| **Xlarge** | **Load Time, ms** | 520 | 460 | 495 | **492** |
| | **Latency, ms** | 310 | 295 | 300 | **302** |

**Table 5.14**: Jmeter benchmarking results for Amazon EC2 instances

As can be seen from the tables for Small and Medium machine types, the overall networking performance of the Medium type is much better. This fact can even be noticed while browsing the WordPress blog using a web browses - the response time is much faster when a larger instance is used.

Amazon EC2 offers different of bandwidth with different virtual instances. Smaller machines get less bandwidth than the bigger ones. Unfortunately the provider doesn't specify the exact performance data for the network connection for the m1 instance type that was used in this research (the most common general purpose instance type on Amazon EC2), it only mentions that the Small instance comes with the "Low" network performance, Medium and Large types with "Moderate" network performance, and the Xlarge type with "High" network performance.

As expected, the network performance data for Large and Xlarge instance types is better than

for smaller ones. The difference between Small and Medium types is the most significant. This can be explained by the fact that Small instance is intended mostly for computational purposes, but not for network operations. It is not recommended to implement online shops even for testing purposes and blogs or websites, which will be accessed by a large number of users on this type of instances [86].

The two graphs below provide visual representation of network performance benchmarking.

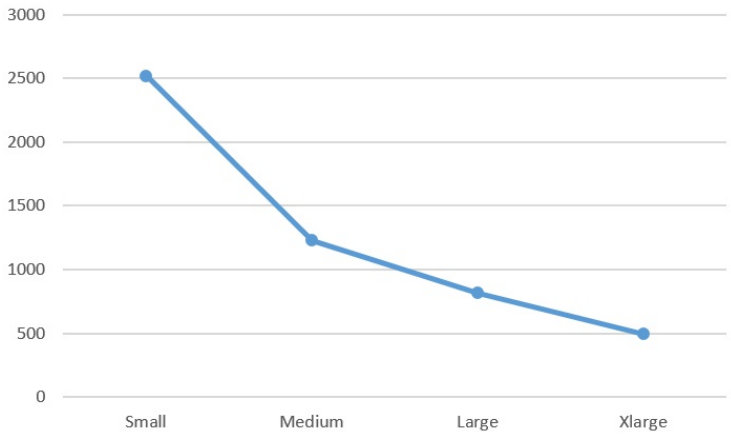


**Figure 5.6**: JMeter Load Time in milliseconds for Amazon EC2 instances

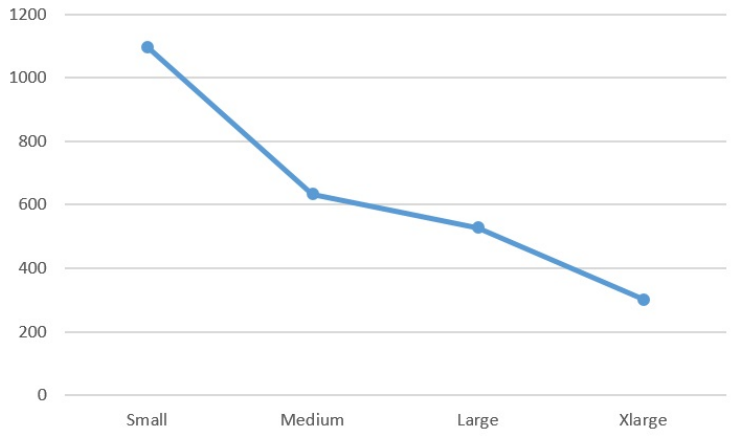


**Figure 5.7**: JMeter Latency in milliseconds for Amazon EC2 instances

There is a significant difference in network performance between Medium and Large virtual instance types, however, as specified on the Amazon EC2 website, they both have "Moderate" network performance, so no difference is expected. This fact of course will not disappoint users of Large instances, but the characteristics of network performance, specified by Amazon, turns out to be imprecise. The latency values, however, don't differ significantly between these two

types, but the difference still exists.

## Google GCE

In this part the JMeter benchmarking results for Google GCE are presented. The structure is similar to the previous part, where network performance of Amazon EC2 virtual machines were discussed. First, the table presents the benchmarking data of four types of virtual instances in Google cloud. Then, two charts show visualization of that data. Finally, at the end of this section, network performance of both Amazon EC2 and Google GCE is compared.

The table below presents the benchmarking results of network performance of different Google GCE instance types.

| | | Run 1 | Run 2 | Run 3 | Average |
|---|---|---|---|---|---|
| **Small** | **Load Time, ms** | 180 | 200 | 210 | **197** |
| | **Latency, ms** | 135 | 120 | 140 | **132** |
| **Medium** | **Load Time, ms** | 170 | 250 | 240 | **220** |
| | **Latency, ms** | 125 | 150 | 130 | **135** |
| **Large** | **Load Time, ms** | 165 | 170 | 185 | **173** |
| | **Latency, ms** | 90 | 105 | 100 | **98** |
| **Xlarge** | **Load Time, ms** | 160 | 190 | 165 | **172** |
| | **Latency, ms** | 110 | 80 | 105 | **98** |

**Table 5.15**: Jmeter benchmarking results for Google GCE instances

The overall networking performance of GCE instances is significantly higher than of Amazon EC2. Both page load time and latency are much better compared to Amazon. The interesting fact is that the performance of virtual instances of different types is similar and the values don't change significantly. However, a small change can be noticed between Small plus Medium types and Large plus Xlarge types - large instances have slightly better performance. Google Compute Engine does not specify the instance networking performance data for its general purpose instances. But as has been discovered, all instances have close networking performance parameters values which do not depend on the instance type.

From the above figures it can be seen that the GCE instances belong to two classes of instances based on their networking performance. Small and Medium instance types have more or less similar performance, and Large and Xlarge have identical network performance values.

The next figures display comparison between Amazon EC2 and Google GCE virtual machines from the perspective of network performance. With the Amazon EC2 network performance graph on the background, the difference between different instance types of Google GCE instances almost disappears.

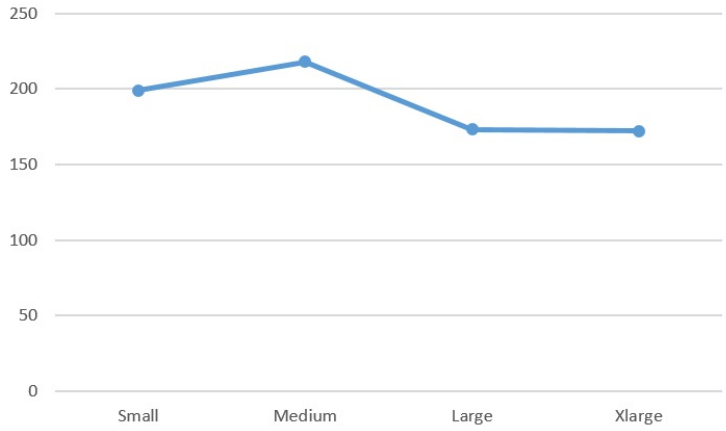


**Figure 5.8**: JMeter Load Time in milliseconds for Google GCE instances

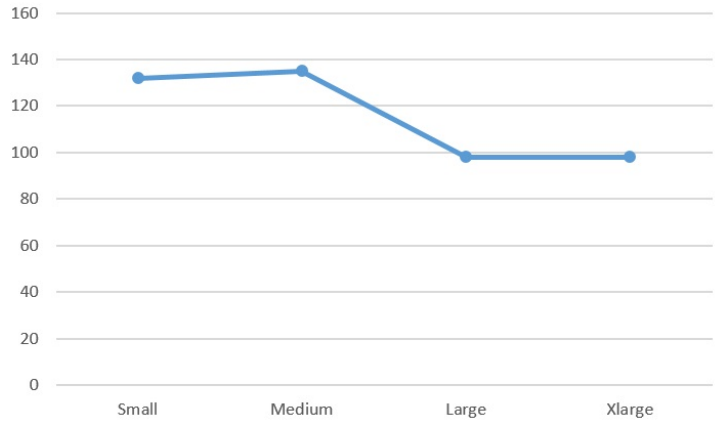


**Figure 5.9**: JMeter Latency in milliseconds for Google GCE instances

At the first look, these results, displayed on the next two figures, can seem a bit confusing, but such a big difference in page load time and latency between Amazon EC2 and Google GCE virtual machines has several reasons. First, as has been indicated by the Iperf benchmark, Google VMs have much better network performance (the bandwidth is two times higher in case of Google). Another reason is that there are artificial restrictions on virtual machines of smaller types in Amazon cloud. Virtual machines of type Small have the most severe restrictions. Also, many researches mention [3], [8], [44], that virtual instances of type Small have very limited functionality, and they are intended mostly for learning or setting-up purposes rather than for deploying some applications that require high performance.

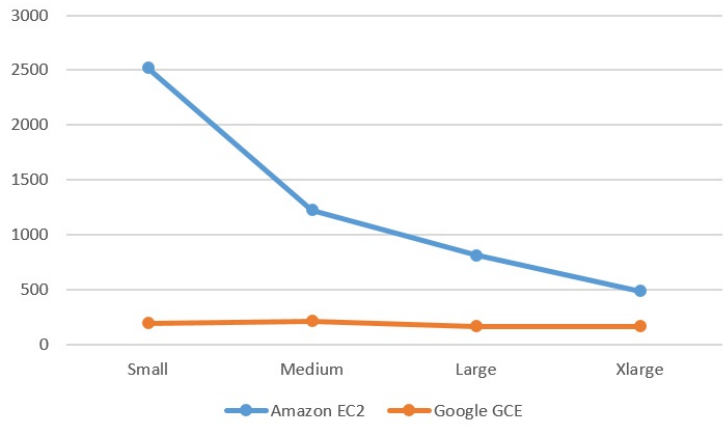


**Figure 5.10**: Load Time in milliseconds comparison between Amazon EC2 and Google GCE instances

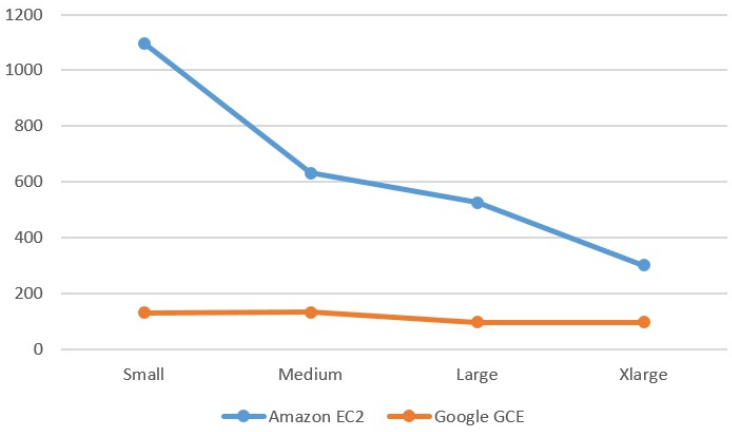


**Figure 5.11**: Latency in milliseconds comparison between Amazon EC2 and Google GCE instances

The following table presents the price-performance ratio for different virtual machine types, used in this work, based on the results of the JMeter latency benchmark. Since the higher latency implies worse performance, the inverse latency value was divided by the VM price. The results are displayed in computational values. These values are not related to the computational values, used in previous sections. According to these results, VMs in Google clouds have significantly higher ratio compared to Amazon VMs.

| | Amazon EC2 | Google GCE |
|---|---|---|
| **Small** | 0.012 | 0.176 |
| **Medium** | 0.010 | 0.086 |
| **Large** | 0.007 | 0.059 |
| **Xlarge** | 0.005 | 0.028 |

**Table 5.16**: Price-performance ratio for different VM types from Amazon EC2 and Google GCE based on JMeter latency benchmarking, in computational values

## 5.4 RUBiS Benchmarking

This section contains benchmarking results and discussion for RUBiS, performed on Amazon and Google virtual instances.

### Amazon EC2

The optimal number of users for RUBiS benchmark is 5000 - 8000. There were two series of experiments that represent the output when configuring both values.

The pure outcome of RUBiS benchmark is a "requests per second" characteristic, which represents how many requests per second one virtual user on a client instance can remotely make to a server instance. It worth mentioning, that this characteristic does not change significantly in different instance types, but the CPU load changes a lot, which is expected. Also, the percentage of CPU, "stolen" by the virtualizer, changes sharply depending on the instance type (or more precisely, the current CPU load level), which might seem unusual.

“id” is the "Idle" field from the “top” command in Linux;

“st” is the “Steal Time” CPU field from the “top” command in Linux;

CPU load is calculated by subtracting the "Idle" value from 100 per cent.

| | Small | Medium | Large | Xlarge |
|---|---|---|---|---|
| **id, %** | 0 | 28 | 53 | 80 |
| **st, %** | 45 | 0.04 | 0.14 | 0.07 |
| **CPU load, %** | 100 | 72 | 47 | 20 |
| **req/s** | 44 | 49 | 50 | 51 |

**Table 5.17**: RUBiS Benchmark Results for Amazon. 8000 Users

| | Small | Medium | Large | Xlarge |
|---|---|---|---|---|
| **id, %** | 0 | 50 | 79 | 92 |
| **st, %** | 46 | 0.04 | 0 | 0.01 |
| **CPU load, %** | 100 | 50 | 21 | 8 |
| **req/s** | 46 | 50 | 51 | 52 |

**Table 5.18**: RUBiS Benchmark Results for Amazon. 5000 Users

All the data in the following two tables has been collected using the following approach: three instances of each type have been created. Each of them was benchmarked three times. Therefore,

Each measured value in the table is a grand-average of nine measurements.

| | New | | | | New | | | | New | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **1** | **2** | **3** | **avg1** | **1** | **2** | **3** | **avg2** | **1** | **2** | **3** | **avg3** | **AVG** |
| **Idle** | 53 | 56 | 55 | **54.67** | 55 | 52 | 54 | **53.67** | 54 | 50 | 51 | **51.67** | **53.3** |
| **Steal** | 0.3 | 0.2 | 0.2 | **0.23** | 0 | 0 | 0.1 | **0.03** | 0.1 | 0.2 | 0.2 | **0.16** | **0.14** |
| **req/s** | 49 | 48 | 49 | **48.67** | 51 | 53 | 51 | **51.67** | 49 | 48 | 51 | **49.33** | **49.89** |

**Table 5.19**: Example of Data Calculation for Instance Type Large, 8000 Users

As can be seen from the tables, the benchmark characteristic requests per second, "req/s", only slightly changes depending on the virtual instance type. In fact, this value can drop if the connection between client and server instances is unexpectedly slow or is being disrupted. Another reason for the value drop can be misconfiguration of LAMP instance (Linux, Apache, MySQL, PHP) or RUBiS benchmark, which still allows benchmark to function, but its results are unreliable (for example, if some RUBiS configuration scripts have not been run, or were interrupted, and there is not enough data in the database to function properly). The highest number of requests per second, which was detected during all series of experiments in this work, is 56.

The trend is similar in both cases (5000 and 8000 users), and RUBiS characteristic req/s is higher on larger machine types.

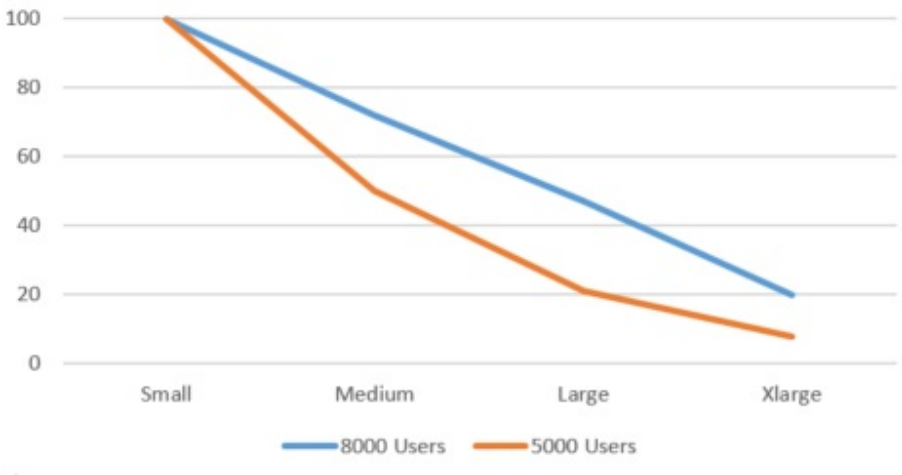


**Figure 5.12**: CPU load for different instance types and 5000 or 8000 users in Amazon

The dependency of the number of users simulated in RUBiS on the CPU load for different instance types is represented in the figure. The graph indicates the common trend that CPU load is lower on more powerful instances. Also, it can be seen, that the larger number of users implies higher CPU load. For the VM of type Small the load is about 100 per cent, but it is not overwhelming - after a series of attempts during the RUBiS benchmark configuration and calibration procedure, the number of users was selected to correspond to 100 per cent of CPU load on the instance of type Small.

Another interesting observation is the "stolen" CPU capacity. From tables 5.17 and 5.18, it can be seen that for the instance of type Small the amount of "stolen" CPU is about 50 per cent, while

on larger instances it is close to zero. It turns out that if the CPU load is not high enough (not close to 100 per cent), then the "top" command in Linux can not detect if some CPU is "stolen" by virtualizer [43], [54]. In this experiment the CPU load of the Small instances was close to 100 per cent, so the "top" command detected that some CPU capacity (45 per cent) is being "stolen" by the virtualizer for some other processes. Reasons for that are explained in previous chapter. Virtual instances of larger sizes were not fully loaded during the experiment, and therefore the amount of "stolen" CPU was close to zero. Another reason for such a high level of a CPU "steal time" for instances of type Small compared to another instance sizes is the fact that this type of virtual instances is the most popular on Amazon EC2 [44]. Since there are many similar instances running at the same time, and probably on the same hardware in Amazon computational centers, virtualizer might not be able to assign CPU capacities to all virtual instances without loosing computational power.

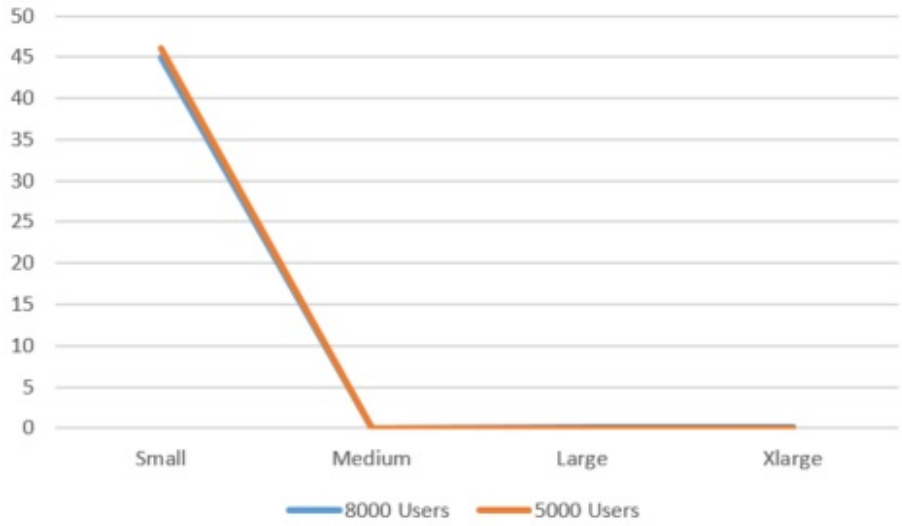


**Figure 5.13**: The amount of "stolen" CPU performance for different instance types and 5000 or 8000 users in Amazon EC2

## Google GCE

In this part the results of series of experiments with Google GCE virtual instances and RUBiS benchmark are discussed.

The number of virtual users for RUBiS benchmark was taken the same as for Amazon EC2, 5000 and 8000 users. This decision was made in order to keep experiment conditions for both cloud providers the same. Later it turned out that the instance of type Small from GCE is loaded also for about 100 per cent with this user amount, which is one of the requirements for the experiment.

The experiment structure is similar to one used for Amazon EC2 earlier in this section, and the data being gathered is the following:

"id" is the "Idle" field from the "top" command in Linux;

"st" is the "Steal Time" CPU field from the "top" command in Linux;

CPU load is calculated by subtracting the "Idle" value from 100 per cent.

Two tables below represent the performance of Google GCE virtual instances with RUBiS benchmark configured for 5000 and 8000 users.

| | Small | Medium | Large | Xlarge |
|---|---|---|---|---|
| **Idle** | 0 | 29.8 | 54.5 | 76 |
| **Steal Time** | 0.12 | 0 | 0 | 0 |
| **CPU Load** | 100 | 70.2 | 45.5 | 24 |
| **req/s** | 36 | 39 | 41 | 41 |

**Table 5.20**: RUBiS Benchmark Results for Google. 8000 Users

| | Small | Medium | Large | Xlarge |
|---|---|---|---|---|
| **Idle** | 0 | 34 | 61.7 | 81.6 |
| **Steal Time** | 0.09 | 0 | 0 | 0 |
| **CPU Load** | 100 | 66 | 37.3 | 18.4 |
| **req/s** | 37 | 39 | 40 | 41 |

**Table 5.21**: RUBiS Benchmark Results for Google. 5000 Users

If we compare the requests per second characteristic from both Amazon and Google, we will see that instances from Google have on average 10 req/s less performance than similar instances from Amazon. This result can be explained by the fact that Google virtual instances have about 12 per cent less RAM assigned to them. But RAM can not be solely responsible for a 25 per cent decrease of the req/s value. Another factors can be differences of hardware used in computing centers at Amazon and Google, different virtualizer settings, computing center load, etc.

However we can notice that virtual instances from Google almost don't have any CPU capacity "stolen" by virtualizer for other purposes. The "Steal time" for Small VM type value is about 1 per cent, while on more powerful machines it is zero. What could be the reasons for that? There are two explanations. First, the virtualizer configuration, which does not allow any "stealing" of CPU power under any circumstances. And secondly, the particular computing center or hardware on which the virtual machine resides, is not under any heavy load at the moment.

The next table presents the price-performance ratio for different virtual machine types, used in this work, based on the results of the RUBiS benchmark for 5000 simulated users. Since the higher req/s value implies better performance, the req/s value was divided by the VM price. The results are displayed in computational values. These values are not related to the computational values, used in previous sections. According to the presented data, VMs in Google clouds have slightly higher ratio than Amazon VMs.

| | Amazon EC2 | Google GCE |
|---|---|---|
| **Small** | 597 | 880 |
| **Medium** | 318 | 464 |
| **Large** | 162 | 238 |
| **Xlarge** | 83 | 122 |

**Table 5.22**: Price-performance ratio for different VM types from Amazon EC2 and Google GCE based on RUBiS benchmarking for 5000 simulated users, in computational values

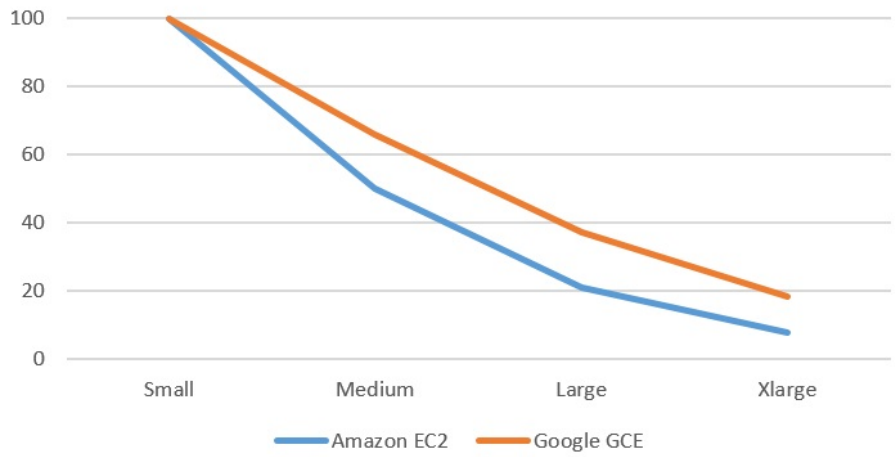


**Figure 5.14**: CPU load for different instance types and 5000 users in Amazon EC2 and Google GCE

After reviewing the performance of the virtual machines, CPU load and the "stolen" CPU, the question arises: Why do cloud providers don't configure virtualizers in such a way that they don't allow "stealing" or borrowing resources from a virtual instance? Similar to internet service providers, who's customers share bandwidth of an uplink channel, it would be irrational for cloud providers to assign computing capacities to their customers on a "one-to-one" basis, i.e. when providers offer only the amount of computing power that they have on site. Since hardly ever all hardware is fully loaded, it would be a huge waste of resources. Instead, providers use virtualizers to automatically detect the hardware that is currently not loaded, and create a virtual instance on it. In this case, hardware is often reused, which increases the overall hardware load, but allows allocating more virtual instances than with "one-to-one" mapping. As a downside of this approach, when a computing center experiences heavy loads, situations can occur, when there is not enough resources for a virtual machine to run, and the virtualizer has to "steal" computing capacity from other customers.

Such cases are taken care by a cloud provider, and the total amount of "stolen" processing power should not increase to the level that would significantly cut the performance of a virtual instance, that is specified in a service level agreement. And the 45 per cent of "stolen" CPU speed means in fact that this amount of CPU speed, that was supposed to be consumed by this virtual machine, has been assigned to another process. But at the same time in most cases a certain amount of CPU power from some another process has been assigned to this virtual machine to compensate the loss. This amount can not be estimated with the "top" Linux command. All these

processes of course influence the overall performance of a virtual machine, because the best case scenario would be to run the VM on one hardware without any other neighboring processes that require a lot of resources. These processes are performed by the virtualizer in computing centers of cloud providers.

## 5.5 Performance Correspondence Between Cloud-Based VMs and Physical Machines

Most of the articles about prediction of the actual performance of cloud-based virtual machines, especially the early ones, point out that the level at which cloud providers keep their services, is not suitable for tasks that require performance stability, because there was little correspondence between specified characteristics on a provider's website and real performance [58], [59], [68].

The two tables below give a performance comparison between virtual machines and similar physical machines. The data for physical machines was collected from the openbenchmarking.org website[1], which has benchmarking data for a wide variety of benchmarks for a system with almost any hardware components.

| | **Physical Machine** | **Virtual Instance** | **Difference** |
|---|---|---|---|
| | **Gzip** | | |
| **Small** | 56.8 s | 55.5 s | -1.3 s |
| **Large** | 25.7 s | 25.2 s | -0.5 s |
| | **C-Ray** | | |
| **Small** | 362.1 s | 367 s | +4.9 s |
| **Large** | 217.4 s | 211.5 s | -5.9 s |
| | **Bork** | | |
| **Small** | 69.7 s | 71.5 s | +1.8 s |
| **Large** | 29.4 s | 28.6 s | -0.8 s |
| | **Ramspeed** | | |
| **Small** | 2995.44 MB/s | 2977.68 MB/s | -17.76 MB/s |
| **Large** | 15370.25 MB/s | 15887.35 MB/s | +517.10 MB/s |

**Table 5.23**: Performance comparison between a virtual machine and similar physical machine. Data for Amazon EC2

The performance difference in the above tables doesn't exceed the error of 5 per cent, and this allows us stating that performance, specified on the provider's website corresponds to the actual performance of virtual machines.

It is not possible to confirm the network correspondence, because for the general purpose virtual machines, which were used in this work, providers only specify that a certain VM has either "low", "moderate" or "high" bandwidth, without any exact numbers. But as experimental results indicated, larger VM has higher bandwidth. However, in case of Amazon EC2, provider specifies that machines of type Medium and Large both have "moderate" bandwidth, and the user can assume that it is equal on both instances, but in fact, Medium instance type has bandwidth of 866 MBit/s, and Large has 912 MBit/s, and this difference (46 MBit/s) exceeds 5 per cent.

[1] https://openbenchmarking.org/

| | Physical Machine | Virtual Instance | Difference |
|---|---|---|---|
| | **Gzip** | | |
| **Small** | 56.6 s | 54.6 s | -2 s |
| **Large** | 17.2 s | 16.4 s | -0.8 s |
| | **C-Ray** | | |
| **Small** | 366 s | 355.4 s | -10.6 s |
| **Large** | 139.1 s | 135.8 s | -3.3 s |
| | **Bork** | | |
| **Small** | 117.2 s | 119.8 s | +2.6 s |
| **Large** | 26.4 s | 27.8 s | +1.4 s |
| | **Ramspeed** | | |
| **Small** | 5711.62 MB/s | 5538.52 MB/s | -173.1 MB/s |
| **Large** | 9535.74 MB/s | 9223.63 MB/s | -312.11 MB/s |

**Table 5.24**: Performance comparison between a virtual machine and similar physical machine. Data for Google GCE

General purpose instances from these two IaaS providers don't have exactly the same specifications. Google uses Intel Xeon 2.5 processors instead of Intel Xeon 2.0 which is used in Amazon. Also, Google has uneven amount of RAM configured for its instances. More detailed information about hardware can be found in chapter 4. But both providers apply the term "general purpose" for these machines, and that is the reason why they were considered similar from the end user perspective.

# Chapter 6

# Discussion

Based on the experimental results, it is not easy to determine the cloud service that has better performance and which could be a "winner" in this comparison. Both Amazon EC2 and Google GCE have strong and week areas. According to the data gathered by all the benchmarks in this work, it can be concluded that virtual machines of all types, running in Google cloud, have better CPU and network performance, and better RAM performance on the instance of type Small. On the contrary, according to the results of RUBiS benchmark, Amazon EC2 instances have better overall system performance when running hosted OLTP application, such as an eBay-like website, installed and configured for benchmarking OLTP performance. Also, Amazon's virtual machines of type Large (8 GB RAM) and Xlarge (16 GB RAM) have better RAM performance compared to similar Google VMs. As for the network, Google is a clear winner in this category, having both bandwidth and latency values higher than Amazon VMs. More detailed comparison and explanation can be found in chapter 5 - Results and Data Analysis.

The parts that presented the price-performance ratio of the virtual machines, tested in this work, demonstrated that if taking into account both VM performance and rental price, then in most cases Google GCE VMs are preferred over machines in Amazon cloud. Google has cheaper offerings, and this fact can attract a large number of customers, especially those, who start exploring IaaS clouds and don't wish to invest much money at first time. But it would be wrong to assume that Google virtual machines have low performance since they are cheaper. In most scenarios their performance is better or similar to the performance of Amazon VMs.

Another important factor that has to be taken into account is the price of virtual instances. Google GCE machines are cheaper than similar Amazon EC2 machines. Combined with the performance benchmarking results, that indicated that Google performs in certain tasks better than Amazon, and in those where it performs worse, the difference is insignificant, Google GCE offerings might be more attractive to customers, unless some specific requirements are set, for example if high RAM performance is a key.

There were some interesting findings in the results of benchmarking both cloud providers. The most surprising finding was that the network performance of the Google GCE virtual machines is much better than at Amazon. The obvious explanation of that fact is the bandwidth, which is about two times higher between VMs in Google cloud. Another reason could be the underlying network infrastructure that can limit the speed with which the data packets are transferred (used protocol HTTP, port 80), for example ACLs that filter packets based on port numbers, different security groups and policies, used in Amazon cloud. All this factors can slow down the traffic between hosts on a network and increase latency.

An important fact to mention is the performance data determined by different benchmarks for the VMs of type Small in both Amazon and Google clouds. Even though the performance of such instances has been stable at different times and also after re-creation of the instance, the benchmarking data might seem to be a bit confusing. For example, Google VM of type Small has

better RAM performance compared to similar VM in Amazon cloud, but the Amazon's VM has a bit more RAM (2 GB instead of 1750 MB), and is expected to perform better. The answer to this issue is the virtualization rules, deployed by Amazon EC2. As was noticed while benchmarking CPU performance, the amount of "stolen" CPU for Small VM (CPU, reassigned to some other processes by the virtualizer) is close to 50 per cent, compared to 1-2 per cent in Google cloud. This fact has been confirmed by a number of researches [3], [8], [44], [56] and it implies that this VM type in Amazon cloud is built together from the unused powers on different physical components in a data center. Therefore the actual performance of such virtual machines is less than expected. Moreover, it is not recommended to deploy high risk applications on the VMs of type Small or use these virtual instances for any critical processes due to possible performance variations.

Another interesting finding is the generally high performance of Google GCE virtual machines. Taking into account the price of Google VMs compared to the price of Amazon VMs, customers would expect lower performance of VMs in Google cloud. But as the experiments have indicated, virtual machines in Google GCE have better or similar performance compared to Amazon EC2, except of the OLTP performance that characterizes the system's combined performance (interoperability).

As for the practical application of the virtual machines, that had been benchmarked in this work, it is also not easy to say, which exact VM suits certain task better. But after having tested the actual performance of general-purpose machines, it is possible to make conclusions whether one cloud service provider is better than another one from a certain perspective.

**CPU and RAM performance.** Based on the results of Phoronix Test Suite benchmarking, Google GCE virtual machines have significantly better CPU performance. From the figures and tables presented in previous chapter, the difference between Google and Amazon VMs is noticeable. RAM performance is better in Amazon machines. But the combined performance (evaluated by the Bork benchmark) is slightly better on Google virtual machines. As for the computer clusters, Google GCE is a winner too, since it has significantly better network performance, which has been proven by JMeter and Iperf benchmarks, and network is one of the key factors for computer clusters.

**OLTP performance.** For running online transaction protocol (OLTP) applications, all system's components should have similar performance, so that none of them becomes a bottleneck. From the hardware point of view these applications include interactions between CPU, RAM, I/O device and network. An OLTP-based application can be an interactive website, a blog, or a CRM.

After benchmarking virtual machines with the help of RUBiS benchmark, which simulated an OLTP application, similar to eBay auction site, it can be stated that the performance of virtual machines in Amazon cloud is better. All Amazon VMs generated more requests per second (calculated value by RUBiS), and they had smaller CPU load. That means that all system components of Amazon VMs in combination perform better than Google VMs. But the fact that Google machines are cheaper than Amazon's instances, can overweight if price is the main issue.

**Network performance.** One of the possible use cases of virtual machines in cloud could be using it as a network node. A VM can be used as a VPN or proxy server or as a node in a chain of network nodes. Regardless of the purpose, network performance is the key for a virtual machine in this case. All network parameters matter, including bandwidth, jitter and latency. As the results of JMeter and Iperf benchmarking have indicated, the network performance of Google GCE machines is incomparable to Amazon EC2. Google virtual machines have outstanding performance, even VMs of type Small. Since VMs that are intended to be network nodes don't require high CPU or RAM performance, the largest VM type for this purpose is considered to be Medium,

but Amazon applies severe bandwidth restrictions on these machine types, and this is the reason why Amazon EC2 VMs are not competitors to Google GCE in terms of network performance. Moreover, Google virtual machines cost less than Amazon instances. Without any doubt, Google GCE is preferable provider in this category.

# Chapter 7

# Threats to Validity

As any large research, this work has a number of limitations, which are worth mentioning. The major limitation is that all experiments had been conducted in one region - Western Europe. Both Amazon EC2 and Google GCE have data centers in many locations in the world. Virtual instances in each region have different performance data and pricing. Due to this fact, the results of this work can not be applied to any other region than Western Europe. This work has scientific value only for one region, and for people form other regions it can serve for informational purposes or as a guideline how to perform all or a subset of experiments described on virtual machines in another region.

Another important limitation is that cloud technology is developing and changing very fast. It sounds positive, but from the perspective of this work, this is the factor that can make the data presented here outdated much faster than in other areas of science. The experimental part of this work had been conducted between January and May 2015, and the results are valid for virtual instances and services that were available at Amazon EC2 and Google GCE at that time. These two cloud providers are implementing new technologies and configurations all the time, and they tend to update their hardware to newer versions almost every year. This work represents current state of having a virtual machine at Amazon or Google, running on hardware that exists today. And most likely after few years, when the hardware and used technologies will be upgraded and changed, the results of this work will only serve for informational purposes. The clear trend can be noticed after reading the Related Work section: in 2008 the performance, stability and reliability of virtual instances was relatively low, and companies could not rely on cloud IaaS providers. Later, in 2012, there was a major improvement of the quality of IaaS products. And now, 3 years later, the difference in performance between physical machines and virtual instances still exists, but it is almost eliminated. So are the articles written back in 2008 - today they only serve for informational purposes.

Finally, the limitation related to the process of benchmarking is the selection of only two IaaS cloud providers. If more providers would have been selected for benchmarking, it would be possible to get an overview of the larger number of IaaS offerings available today. But since the two providers that were selected (Amazon EC2 and Google GCE) are the main players on the IaaS market, they reflect the general trend which all their competitors follow.

# Chapter 8

# Conclusion

This work describes the experiments performed with a set of virtual machines of two largest IaaS cloud providers on the market - Amazon EC2 and Google GCE. The results of these experiments are intended to help researches, developers and other categories of IaaS cloud users to choose the virtual machine that has the best performance, required for the tasks that the user wishes to perform with it. The performance of rented virtual machines was tested using different benchmarking techniques and tools, such as micro benchmarks for evaluating a system's hardware components performance in isolation, Iperf and JMeter tools for evaluating network performance, and RUBiS platform, which is an OLTP based auction site emulator, similar to eBay, that evaluates the performance of a system as a whole, including the interoperability of all its components.

As the experimental results have indicated, there is no clear winner between the two cloud service providers. Amazon EC2 has better overall system performance, as was proved by the RUBiS benchmark, and also Amazon virtual machines of types Large and Xlarge have better RAM performance compared to Google virtual machines. Google VMs demonstrated higher results in computational performance (CPU speed and CPU plus RAM performance). Furthermore, network performance of Google GCE virtual instances is significantly better than VMs at Amazon EC2. Finally, the price of the similar instances at Amazon EC2 and Google GCE differs a lot. In general, prices for Google virtual machines are about 40 per cent smaller compared to Amazon VMs. The lower price for GCE instances in most cases will be the key factor when choosing between these two providers, because in those categories, where Amazon virtual machines perform better, the difference in performance is not large enough to persuade users to pay more.

## 8.1 Research questions revisited

This section gives answers on two research questions, stated when starting this work.

**RQ1.** *Does the real performance of virtual machines match the characteristics specified by the service provider?*

The cloud-based virtual machines demonstrated almost the same performance compared to the performance of the physical machines with similar architecture and hardware components. In all cases the performance variation didn't exceed 5 per cent, which allows stating that the answer on the first research question is yes. Although there were clear deviations in performance even between two instances of the same type, these deviations were not significant and they didn't affect the average performance value in such a way that the difference between a cloud-based VM and a corresponding physical machine is more that 5 per cent.

**RQ2.** *Do similar virtual machines in Amazon EC2 and Google GCE have similar performance?*

Based on the results of the cloud-based virtual machines, it can be concluded that the answer on the second research question is no. The actual performance data of instances of the same type

in Amazon and Google is different. And in some cases this difference is compelling. As the performance evaluation has indicated, Google VMs have better computational and network capacity, while Amazon VMs have better overall system performance.

The results of this work are only suitable for the region of Western Europe for both Amazon EC2 and Google GCE, since all virtual instances had been created and experiments had been conducted in this computational region. Therefore the results of similar experiments at another region will differ from the results presented in this work. Furthermore, the change of hardware used in computing centers of cloud service providers, virtualization techniques and pricing plans, will also affect the results of similar experiments, performed after such changes. Therefore, the results of this study are only valid when all the mentioned parameters stay the same as they were in January-May 2015, when this work had been performed. But nevertheless, this work can serve as a guideline for future researches on how to conduct similar experiments, install and configure system performance evaluation tools and perform benchmarking of cloud-based services.

## 8.2 Outlook

The future research that could be a possible continuation of this work, could include extension of benchmarking tools being used. In particular, it would be beneficial to deploy a second benchmark, that can evaluate the overall system performance and components interoperability, similar to RUBiS. This would allow researchers to compare results of two independent benchmarks, that perform similar evaluation. Another direction of continuing this study is to get a community that is specialized in cloud computing and cloud benchmarking involved in this research. In particular, a group of people could evaluate performance of most popular virtual machines after any change in hardware components or a virtualization mechanism, deployed by the cloud service provider, and then gather this data in a table or any other form that would allow easy comparison of different service providers and products they offer. Amazon EC2 and Google GCE infrastructure allows creating images of virtual machines, so they could be easily recreated at any time, using any VM type available at the moment, without any additional configuration. There are already some communities that perform benchmarking of different cloud services (the most popular one is CloudHarmony[1]), but current communities don't have benchmarking results for all available VM types, even from the two major IaaS cloud providers. There are specialized forums, dedicated to either Amazon EC2 and its products, or to Google GCE, where users of these IaaS clouds share their experience and help others to resolve problems that new users usually face when they start using cloud services. If there could be possible to perform benchmarking of cloud-based VMs and post the results there, that would significantly simplify the process of choosing the best offering for customers.

When cloud-based virtual machines become more popular, their performance becomes more stable, cloud services become more secure and transparent, and their price gets lower, then many businesses will draw attention to them as the way of saving money and decreasing IT configuration complexity. This would mean the start of the era of widespread cloud deployment. So far we can see that cloud technologies are moving towards this, cloud providers are constantly improving their products, and prices for cloud resources are decreasing. Researches and developers, who work in the area of cloud analysis and performance evaluation, help to bring closer that era by sharing their knowledge with partners and educating future cloud service users.

This work provides the results of benchmarking of two major IaaS cloud providers, and it also draws attention to the fact that it is important to know the actual performance of cloud services, before selecting one or another provider to start a virtual machine in its cloud. This performance

[1] https://cloudharmony.com/

data in combination with the analysis of VM prices assists in making a right choice when selecting a virtual machine for particular purposes.

# Bibliography


[1] P.Leitner, J.Cito. "Patterns in the Chaos-a Study of Performance Variation and Predictability in Public IaaS Clouds." arXiv preprint arXiv:1411.2429 (2014).

[2] J.Scheuner, et al. "Cloud WorkBench-Infrastructure-as-Code Based Cloud Benchmarking." arXiv preprint arXiv:1408.4565 (2014).

[3] A.Lenk, et al. "What are you paying for? performance benchmarking for infrastructure-as-a-service offerings." Cloud Computing (CLOUD), 2011 IEEE International Conference.

[4] M.Cunha, N.Mendonça, A.Sampaio. "A declarative environment for automatic performance evaluation in IaaS clouds." Cloud Computing (CLOUD), 2013 IEEE Sixth International Conference on. IEEE, 2013.

[5] S.Ostermann, et al. "A performance analysis of EC2 cloud computing services for scientific computing." Cloud computing. Springer Berlin Heidelberg, 2010. 115-131.

[6] C.Curino, et al. "Benchmarking OLTP/Web databases in the cloud: the OLTP-bench framework." Proceedings of the fourth international workshop on Cloud data management. ACM, 2012.

[7] Z.Li, L.O'Brien, Z.He. "CEEM: A practical methodology for Cloud services evaluation." Services (SERVICES), 2013 IEEE Ninth World Congress on. IEEE, 2013.

[8] J.Dejun, G.Pierre, C.Chi-Hung. "EC2 performance analysis for resource provisioning of service-oriented applications." Service-Oriented Computing. ICSOC/ServiceWave 2009 Workshops. Springer Berlin Heidelberg, 2010.

[9] D.Jayasinghe, et al. "Expertus: A generator approach to automate performance testing in IaaS clouds." Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on. IEEE, 2012.

[10] J.O'loughlin, L.Gillam. "Towards Performance Prediction for Public Infrastructure Clouds: An EC2 Case Study." Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on. Vol. 1. IEEE, 2013.

[11] A.Borhani, et al. "WPress: An Application-Driven Performance Benchmark for Cloud-Based Virtual Machines." Enterprise Distributed Object Computing Conference (EDOC), 2014 IEEE 18th International. IEEE, 2014.

[12] L.Kelly. Tips for choosing a global cloud infrastructure provider. Computer Weekly, 2014. http://www.computerweekly.com/feature/Tips-for-choosing-a-global-cloud-infrastructure-provider

[13] D.Sullivan. IaaS Provider's List: Comparison and Guide. Tom's IT Pro. http://www.tomsitpro.com/articles/iaas-providers,1-1560.html

[14] B.Butler, "PaaS Primer: What is platform as a service and why does it matter?" Network World, February 11, 2013.

[15] L.Vaquero, et al. "A break in the clouds: towards a cloud definition." ACM SIGCOMM Computer Communication Review 39.1 (2008): 50-55.

[16] J.Barr. "Amazon EBS (Elastic Block Store) - Bring Us Your Data". Amazon Web Services Blog. 2008.

[17] C.Brooks. Amazon's early efforts at cloud computing? Partly accidental. 2010. http://itknowledgeexchange.techtarget.com/cloud-computing/amazons-early-efforts-at-cloud-computing-partly-accidental/

[18] Amazon EC2 details. http://aws.amazon.com/ec2/details/

[19] D.Harris. "What google compute engine means for cloud computing." GigaOM-Tech News, Analysis and Trends. http://gigaom.com/cloud/what-google-compute-engine-means-for-cloud-computing (2012).

[20] Google Compute Engine. Details. https://cloud.google.com/compute/

[21] R.Prodan, M.Sperk, S.Ostermann. "Evaluating high-performance computing on google app engine." Software, IEEE 29.2 (2012): 52-58.

[22] C.Babcock. "Google Wins In Amazon Cloud Price Battle". InformationWeek, 2014. http://www.informationweek.com/cloud/infrastructure-as-a-service/google-wins-in-amazon-cloud-price-battle/d/d-id/1141560

[23] Google Compute Engine pricing. https://cloud.google.com/compute/pricing

[24] P.Kyrö. "Revising the concept and forms of benchmarking." Benchmarking: An International Journal 10.3 (2003): 210-225.

[25] L.Gillam, et al. "Fair benchmarking for cloud computing systems." Journal of Cloud Computing: Advances, Systems and Applications 2.1 (2013): 6.

[26] V.Vedam, J.Vemulapati. "Demystifying Cloud Benchmarking Paradigm-An in Depth View." Computer Software and Applications Conference (COMPSAC), 2012 IEEE 36th Annual. IEEE, 2012.

[27] A.Iosup, R.Prodan, D.Epema. "Iaas cloud benchmarking: approaches, challenges, and experience." Cloud Computing for Data-Intensive Applications. Springer New York, 2014. 83-104.

[28] P.Baillie. Benchmarking cloud servers: A Cloud Computing Insider's Guide. https://www.cloudsigma.com/benchmarking-cloud-servers-a-cloud-computing-insiders-guide/

[29] P.Baillie. SSDs gaining traction in the cloud. https://www.cloudsigma.com/ssds-gaining-traction-in-the-cloud/

[30] G.Babbling. RUBiS Workload Installation Guide. http://sanifool.com/2012/09/03/rubis-workload-simple-installation-guide/

[31] P.Leitner. Fixing the RUBiS Benchmark for Modern Linux Environments. http://wp.ifi.uzh.ch/leitner/?p=340

[32] E.Halili. Apache JMeter: A practical beginner's guide to automated testing and performance measurement for your websites. Packt Publishing Ltd, 2008.

[33] D.Nevedrov. "Using JMeter to Performance Test Web Services." Published on dev2dev (http://dev2dev.bea.com/) (2006).

[34] B.Kurniawan. JMeter is a Java. "Using JMeter." (2003).

[35] A.Borhani, et al. "WPress: An Application-Driven Performance Benchmark For Cloud Virtual Machines." IEEE EDOC 2014.

[36] B.Sotomayor, et al. "Capacity leasing in cloud systems using the Opennebula engine." Workshop on Cloud Computing and its Applications. Vol. 3. 2008.

[37] J.Ellingwood. "How To Install Wordpress on Ubuntu 14.04", 2014. https://www.digitalocean.com/community/tutorials/how-to-install-wordpress-on-ubuntu-14-04

[38] G.Boisvert. "Cloud Service Benchmarks", 2014. http://www.sherweb.com/blog/cloud-servers-benchmarks-c-ray/

[39] M.Christen. "Ray tracing on GPU." Master's thesis, Univ. of Applied Sciences Basel (FHBB), Jan 19 (2005).

[40] B.Ford, S.Susarla. "CPU inheritance scheduling." OSDI. Vol. 96. No. 22. 1996.

[41] J.Hwang, et al. "A component-based performance comparison of four hypervisors." Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on. IEEE, 2013.

[42] A.Tirumala, T.Dunigan, L.Cottrell. "Measuring end-to-end bandwidth with Iperf using Web100." Presented at. No. SLAC-PUB-9733. 2003.

[43] A.Iosup, R.Prodan, D.Epema. Iaas cloud benchmarking: approaches, challenges, and experience. In Cloud Computing for Data-Intensive Applications (pp. 83-104). New York, 2014.

[44] L.Iacono, J.Vázquez-Poletti, C.Garino,, I.Llorente. A Model to Calculate Amazon EC2 Instance Performance in Frost Prediction Applications. In High Performance Computing (pp. 68-82). Berlin, Heidelberg, 2014.

[45] E.Walker. "Benchmarking Amazon EC2 for high-performance scientific computing." The magazine of USENIX And SAGE 33, no. 5 (2008): 18-23.

[46] A.Ruiz-Alvarez, M.Humphrey. An automated approach to cloud storage service selection. In Proceedings of the 2nd international workshop on Scientific cloud computing(ScienceCloud '11). ACM, New York, NY, USA, 39-48. DOI=10.1145/1996109.1996117 http://doi.acm.org/10.1145/1996109.1996117

[47] D.Agarwal, S.Prasad. AzureBench: Benchmarking the Storage Services of the Azure Cloud Platform. In Proceedings of the 2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops and PhD Forum (IPDPSW '12). IEEE Computer Society, Washington, DC, USA, 1048-1057. DOI=10.1109/IPDPSW.2012.128 http://dx.doi.org/10.1109/IPDPSW.2012.128

[48] S.Akioka, Y.Muraoka. HPC Benchmarks on Amazon EC2. In Proceedings of the IEEE 24th International Conference on Advanced Information Networking and Applications Workshops (WAINA), pp.1029,1034, 20-23 April 2010. DOI=10.1109/WAINA.2010.166

[49] A.Gültekin, G.Vehbi. Performance evaluation of cloud computing platforms using statistical methods. Comput. Electr. Eng. 40, 5 (July 2014), 1636-1649. DOI=10.1016/j.compeleceng.2014.03.017 http://dx.doi.org/10.1016/j.compeleceng.2014.03.017

[50] S.Barker, P.Shenoy. Empirical evaluation of latency-sensitive application performance in the cloud. In Proceedings of the first annual ACM SIGMM conference on Multimedia systems (MMSys '10). ACM, New York, NY, USA, 35-46. DOI=10.1145/1730836.1730842 http://doi.acm.org/10.1145/1730836.1730842

[51] C.Vecchiola, P.Suraj, and B.Rajkumar. "High-performance cloud computing: A view of scientific applications." Pervasive Systems, Algorithms, and Networks (ISPAN), 2009 10th International Symposium on. IEEE, 2009.

[52] P.Brebner and A.Liu. Performance and cost assessment of cloud services. In Proceedings of the 2010 international conference on Service-oriented computing (ICSOC'10), E. Michael Maximilien, Gustavo Rossi, Soe-Tsyr Yuan, Heiko Ludwig, and Marcelo Fantinato (Eds.). Springer-Verlag, Berlin, Heidelberg, 39-50.

[53] A.Carlyle, S.Harrell, P.Smith. Cost-Effective HPC: The Community or the Cloud? Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on , vol., no., pp.169,176, Nov. 30 2010-Dec. 3 2010 DOI=10.1109/CloudCom.2010.115

[54] D.Cerotti, M.Gribaudo, P.Piazzolla, G.Serazzi. Flexible CPU Provisioning in Clouds: A New Source of Performance Unpredictability, In Proceedings of the Ninth International Conference on Quantitative Evaluation of Systems (QEST), 2012, vol., no., pp.230,237, 17-20 Sept. 2012. doi: 10.1109/QEST.2012.23

[55] C.Curino, D.Difallah, A.Pavlo, P.Cudre-Mauroux. Benchmarking OLTP/web databases in the cloud: the OLTP-bench framework. In Proceedings of the fourth international workshop on Cloud data management (CloudDB '12). ACM, New York, NY, USA, 17-20. DOI=10.1145/2390021.2390025 http://doi.acm.org/10.1145/2390021.2390025

[56] J.Dejun, G.Pierre, C.Chi. EC2 performance analysis for resource provisioning of service-oriented applications. In Proceedings of the 2009 international conference on Service-oriented computing (ICSOC/ServiceWave'09), Asit Dan, and Farouk Toumani (Eds.). Springer-Verlag, Berlin, Heidelberg, 197-207.

[57] R.Expósito, R.Taboada, G.Pardo, C.Xoán, J.Touriño, R.Doallo. Running scientific codes on amazon EC2: a performance analysis of five high-end instances. In Journal of Computer Science Technology, vol. 13, no. 03, 2013

[58] B.Farley, A.Juels, V.Varadarajan, T.Ristenpart, K.Bowers, M.Swift. More for your money: exploiting performance heterogeneity in public clouds. In Proceedings of the Third ACM Symposium on Cloud Computing (SoCC '12). ACM, New York, NY, USA, , Article 20 , 14 pages. DOI=10.1145/2391229.2391249 http://doi.acm.org/10.1145/2391229.2391249

[59] I.Gent, L.Kotthoff. Reliability of Computational Experiments on Virtualised Hardware. ArXiv e-prints Oct 2011; http://arxiv.org/abs/1110.6288

[60] L.Gillam, B.Li, J.O'Loughlin, A.Singh Tomar. Fair Benchmarking for Cloud Computing systems. In Journal of Cloud Computing: Advances, Systems and Applications 2013, 2:6 doi:10.1186/2192-113X-2-6

[61] S.Hazelhurst. Scientific computing using virtual high-performance computing: a case study using the Amazon elastic computing cloud. In Proceedings of the 2008 annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology (SAICSIT '08). ACM, New York, NY, USA, 94-103. DOI=10.1145/1456659.1456671 http://doi.acm.org/10.1145/1456659.1456671

[62] Q.He, S.Zhou, B.Kobler, D.Duffy, and T.McGlynn. Case study for running HPC applications in public clouds. In Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing (HPDC '10). ACM, New York, NY, USA, 395-401. DOI=10.1145/1851476.1851535 http://doi.acm.org/10.1145/1851476.1851535

[63] Z.Hill, M.Humphrey, A quantitative analysis of high performance computing with Amazon's EC2 infrastructure: The death of the local cluster? In Proceedings of GRID. 2009, 26-33.

[64] Z.Hill, J.Li, M.Mao, A.Ruiz-Alvarez, M.Humphrey. Early observations on the performance of Windows Azure. In Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing (HPDC '10). ACM, New York, NY, USA, 367-376. DOI=10.1145/1851476.1851532 http://doi.acm.org/10.1145/1851476.1851532

[65] G.Imre, H.Charaf, L.Lengyel. Performance Analysis of a Java Web Application Running on Amazon EC2. Acta Electrotechnica et Informatica, Vol. 13, No. 4, 2013, 32–39, DOI: 10.15546/aeei-2013-0046

[66] A.Iosup, S.Ostermann, N.Yigitbasi, R.Prodan, T.Fahringer, D.Epema. Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing. IEEE Trans. Parallel Distrib. Syst. 22, 6 (June 2011), 931-945. DOI=10.1109/TPDS.2011.66 http://dx.doi.org/10.1109/TPDS.2011.66

[67] A.Iosup, N.Yigitbasi, D.Epema. On the Performance Variability of Production Cloud Services. In Proceedings of the 2011 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID '11). IEEE Computer Society, Washington, DC, USA, 104-113. DOI=10.1109/CCGrid.2011.22 http://dx.doi.org/10.1109/CCGrid.2011.22

[68] S.Imai, T.Chestna, C.Varela. Accurate Resource Prediction for Hybrid IaaS Clouds Using Workload-Tailored Elastic Compute Units. In Proceedings of the 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing (UCC '13). IEEE Computer Society, Washington, DC, USA, 171-178. DOI=10.1109/UCC.2013.40 http://dx.doi.org/10.1109/UCC.2013.40

[69] K.ackson, L.Ramakrishnan, K.Muriki, S.Canon, S.Cholia, J.Shalf, H.Wasserman, N.Wright. Performance Analysis of High Performance Computing Applications on the Amazon Web Services Cloud. In Proceedings of the 2010 IEEE Second International Conference on Cloud Computing Technology and Science(CLOUDCOM '10). IEEE Computer Society, Washington, DC, USA, 159-168. DOI=10.1109/CloudCom.2010.69 http://dx.doi.org/10.1109/CloudCom.2010.69

[70] D.Kossmann, T.Kraska, S.Loesing. An evaluation of alternative architectures for transaction processing in the cloud. In Proceedings of the 2010 ACM SIGMOD International Conference on Management of data (SIGMOD '10). ACM, New York, NY, USA, 579-590. DOI=10.1145/1807167.1807231 http://doi.acm.org/10.1145/1807167.1807231

[71] G.Kousiouris, G.Giammatteo, A.Evangelinou, N.Galante, E.Kevani, C.Stampoltas, A.Menychtas, A.Kopaneli, K.Ramasamy Balraj, D.Kyriazis, T.Varvarigou, P.Stuer, L.Orue-Echevarria Arrieta. A Multi-Cloud Framework for Measuring and Describing Performance Aspects of Cloud Services Across Different Application Types. In Proceedings of the 2014 Cloud Computing and Services Science Conference (CLOSER).

[72] D.Kumar Krishnappa, E.Lyons, D.Irwin, M.Zink. Network capabilities of cloud services for a real time scientific application. In Proceedings of the 2012 IEEE 37th Conference on Local Computer Networks (LCN 2012) (LCN '12). IEEE Computer Society, Washington, DC, USA, 487-495. DOI=10.1109/LCN.2012.6423665 http://dx.doi.org/10.1109/LCN.2012.6423665

[73] K.LaCurts, S.Deng, A.Goyal, H.Balakrishnan. Choreo: network-aware task placement for cloud applications. In Proceedings of the 2013 conference on Internet measurement conference (IMC '13). ACM, New York, NY, USA, 191-204. DOI=10.1145/2504730.2504744 http://doi.acm.org/10.1145/2504730.2504744

[74] C.Lee. A perspective on scientific cloud computing. In Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing (HPDC '10). ACM, New York, NY, USA, 451-459. DOI=10.1145/1851476.1851542 http://doi.acm.org/10.1145/1851476.1851542

[75] Z.Li, L.O'Brien, R.Ranjan, M.Zhang. Early Observations on Performance of Google Compute Engine for Scientific Computing. In Proceedings of the 2013 IEEE International Conference on Cloud Computing Technology and Science – Volume 01 (CLOUDCOM '13), Vol. 1. IEEE Computer Society, Washington, DC, USA, 1-8. DOI=10.1109/CloudCom.2013.7 http://dx.doi.org/10.1109/CloudCom.2013.7

[76] A.Li, X.Yang, S.Kandula, M.Zhang. CloudCmp: comparing public cloud providers. In Proceedings of the 10th ACM SIGCOMM conference on Internet measurement(IMC '10). ACM, New York, NY, USA, 1-14. DOI=10.1145/1879141.1879143 http://doi.acm.org/10.1145/1879141.1879143

[77] M.Mao, M.Humphrey. A Performance Study on the VM Startup Time in the Cloud. In Proceedings of the 2012 IEEE Fifth International Conference on Cloud Computing (CLOUD '12). IEEE Computer Society, Washington, DC, USA, 423-430. DOI=10.1109/CLOUD.2012.103 http://dx.doi.org/10.1109/CLOUD.2012.103

[78] P.Mehrotra, J.Djomehri, S.Heistand, R.Hood, H.Jin, A.Lazanoff, S.Saini, R.Biswas. Performance evaluation of Amazon EC2 for NASA HPC applications. In Proceedings of the 3rd workshop on Scientific Cloud Computing Date(ScienceCloud '12). ACM, New York, NY, USA, 41-50. DOI=10.1145/2287036.2287045 http://doi.acm.org/10.1145/2287036.2287045

[79] J.Mukherjee, M.Wang, D.Krishnamurthy. Performance Testing Web Applications on the Cloud. In Proceedings of the 2014 IEEE Seventh International Conference on Software Testing, Verification and Validation Workshops (ICSTW), pp.363,369, March 31 2014-April 4 2014. doi: 10.1109/ICSTW.2014.57

[80] J.O'Loughlin, L.Gillam. Performance Evaluation for Cost-Efficient Public Infrastructure Cloud Use. In Proceedings of the 11th International Conference on Economics of Grids, Clouds, Systems and Services (GECON).

[81] S.Ostermann, A.Iosup, N.Yigitbasi, R.Prodan, T.Fahringer, D.Epema. A Performance Analysis of EC2 Cloud Computing Services for Scientific Computing. In Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, Volume 34, 2010, pp 115-131

[82] Z.Ou, H.Zhuang, A.Lukyanenko, J.Nurminen, P.Hui, V.Mazalov, A.Yla-Jaaski. Is the Same Instance Type Created Equal? Exploiting Heterogeneity of Public Clouds, IEEE Transactions on Cloud Computing, vol. 1, no. 2, pp. 201-214, July-December, 2013

[83] Z.Ou, H.Zhuang, J.Nurminen, A.Ylä-Jääski, P.Hui. Exploiting hardware heterogeneity within the same instance type of Amazon EC2. In Proceedings of the 4th USENIX conference on Hot Topics in Cloud

[84] R.Prodan, M.Sperk. Scientific computing with Google App Engine. Future Gener. Comput. Syst. 29, 7 (September 2013), 1851-1859. DOI=10.1016/j.future.2012.12.018. http://dx.doi.org/10.1016/j.future.2012.12.018

[85] Z.Rehman, F.Hussain, O.Hussain, J.Singh. Is There Self-Similarity in Cloud QoS Data?. In Proceedings of the 2013 IEEE 10th International Conference on e-Business Engineering (ICEBE '13). IEEE Computer Society, Washington, DC, USA, 76-81. DOI=10.1109/ICEBE.2013.12 http://dx.doi.org/10.1109/ICEBE.2013.12

[86] J.Schad, J.Dittrich, J.Quiané-Ruiz. Runtime measurements in the cloud: observing, analyzing, and reducing variance. Proc. VLDB Endow. 3, 1-2 (September 2010), 460-471. DOI=10.14778/1920841.1920902 http://dx.doi.org/10.14778/1920841.1920902

[87] R.Tudoran, A.Costan, G.Antoniu, L.Bougé. A performance evaluation of Azure and Nimbus clouds for scientific applications. In Proceedings of the 2nd International Workshop on Cloud Computing Platforms (CloudCP '12). ACM, New York, NY, USA, Article 4 , 6 pages. DOI=10.1145/2168697.2168701 http://doi.acm.org/10.1145/2168697.2168701

[88] V.Varadarajan, T.Kooburat, B.Farley, T.Ristenpart, M.Swift. Resource-freeing attacks: improve your cloud performance (at your neighbor's expense). In Proceedings of the 2012 ACM conference on Computer and communications security(CCS '12). ACM, New York, NY, USA, 281-292. DOI=10.1145/2382196.2382228 http://doi.acm.org/10.1145/2382196.2382228

[89] G.Wang and T.Eugene. The impact of virtualization on network performance of amazon EC2 data center. In Proceedings of the 29th conference on Information communications(INFOCOM'10). IEEE Press, Piscataway, NJ, USA, 1163-1171.

[90] J.Yao, A.Ng, S.Chen, D.Liu, C.Friedrich, S.Nepal. A Performance Evaluation of Public Cloud Using TPC-C. In Service-Oriented Computing – ICSOC 2012 Workshops.

[91] L.Zhao, A.Liu, J.Keung. Evaluating Cloud Platform Architecture with the CARE Framework. In Proceedings of the 2010 Asia Pacific Software Engineering Conference(APSEC '10). IEEE Computer Society, Washington, DC, USA, 60-69. DOI=10.1109/APSEC.2010.17 http://dx.doi.org/10.1109/APSEC.2010.17

[92] R.Tudoran, K.Keahey, P.Riteau, S.Panitkin, G.Antoniu. Evaluating Streaming Strategies for Event Processing Across Infrastructure Clouds, CCGRID, 2014, 2014 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), 2014 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid) 2014, pp. 151-159, doi:10.1109/CCGrid.2014.89

[93] S.Phillips, V.Engen, J.Papay. Snow White Clouds and the Seven Dwarfs. In Proceedings of the 2011 IEEE Third International Conference on Cloud Computing Technology and Science (CLOUDCOM '11). IEEE Computer Society, Washington, DC, USA, 738-745. DOI=10.1109/CloudCom.2011.114 http://dx.doi.org/10.1109/CloudCom.2011.114

[94] Z.Li, L.O'Brien, H.Zhang, R.Cai. Boosting Metrics for Cloud Services Evaluation — The Last Mile of Using Benchmark Suites. In Proceedings of the 2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA '13). IEEE Computer Society, Washington, DC, USA, 381-388. DOI=10.1109/AINA.2013.99 http://dx.doi.org/10.1109/AINA.2013.99

[95] M.Bux, U.Leser. DynamicCloudSim: simulating heterogeneity in computational clouds. In Proceedings of the 2nd ACM SIGMOD Workshop on Scalable Workflow Execution Engines and Technologies (SWEET '13). ACM, New York, NY, USA, , Article 1 , 12 pages. DOI=10.1145/2499896.2499897 http://doi.acm.org/10.1145/2499896.2499897