Sociedade Educacional Três de Maio - SETREM
Tecnologia em Redes de Computadores

ADRIANO VOGEL

SURVEYING THE ROBUSTNESS AND ANALYZING THE PERFORMANCE IMPACT OF OPEN SOURCE INFRASTRUCTURE AS A SERVICE MANAGEMENT TOOLS

Três de Maio

2015

ADRIANO VOGEL

SURVEYING THE ROBUSTNESS AND ANALYZING THE PERFORMANCE IMPACT OF OPEN SOURCE INFRASTRUCTURE AS A SERVICE MANAGEMENT TOOLS

Undergraduate Thesis
Sociedade Educacional Três de Maio
Faculdade Três de Maio
Curso de Tecnologia em Redes de Computadores

Advisor:
M. Sc. Dalvan Griebler

Três de Maio
2015

AGRADECIMENTOS

Este trabalho é dedicado a minha familia por todo o apoio e pela paciência nos momentos que estive ausente para a realização desse estudo.

Um agradecimento especial ao meu orientador Dalvan Griebler, tanto pela oportunidade e confiança para esse trabalho, quanto pelo esforço da orientação e revisão da escrita.

Agradeço também pela equipe do grupo de pesquisa LARCC e projeto HiPerfCloud pelo suporte e oportunidade de discussões.

TERMO DE APROVAÇÃO

ADRIANO VOGEL

SURVEYING THE ROBUSTNESS AND ANALYZING THE PERFORMANCE IMPACT OF OPEN SOURCE INFRASTRUCTURE AS A SERVICE MANAGEMENT TOOLS

Relatório aprovado como requisito parcial para obtenção do título de **Tecnólogo em Redes de Computadores** concedido pela Faculdade de Tecnologia em Redes de Computadores da Sociedade Educacional Três de Maio, pela seguinte Banca examinadora:

Orientador: Prof. Dalvan Griebler, M.Sc.
Faculdade de Tecnologia em Redes de Computadores da SETREM

Drando. Fauzi de Moraes Shubeita
Faculdade de Tecnologia em Redes de Computadores da SETREM

M.Sc. Vinícius da Silveira Serafim
Faculdade de Tecnologia em Redes de Computadores da SETREM

Dr. Claudio Schepke
Universidade Federal do Pampa (UNIPAMPA) - Alegrete

Três de Maio, 07 de agosto de 2015.

ABSTRACT

Cloud computing is an alternative to offer computational resources for users link different technologies (virtualization, grid computing, cluster and parallel processing). Firstly, public cloud providers started offering services and computational resources over the Internet. Meanwhile, the private clouds began to deploy solutions inside the corporations domain. This cloud type becomes suitable because it increases the security and flexibility and avoids a third party dependency. Currently, there are several IaaS open source cloud solutions available to build a public or private cloud. However, their robustness is still unknown as well as the performance impacts. In this thesis, the goal was to evaluate these IaaS tools, concerning the features and support for robustness (flexibility and resilience). Also, a survey state of the art for performance on a cloud was conducted to highlight cloud's challenges and potential solutions for reducing its performance overhead. The results demonstrated that the open source cloud IaaS solutions can provide high robustness levels and are suitable for enterprise applications. Therefore, the most flexible IaaS tool is the CloudStack as well as the OpenStack is the most resilient. Additionally, an equation to find the reliability rates for experiments on the cloud was created due to the performance variation problem found in the survey.

Keywords: Cloud Computing, Infrastructure as a Service, Survey, Performance Analysis, Computer Networks.

ABSTRACT

A computação em nuvem surgiu como uma alternativa para oferecer recursos computacionais através da utilização de tecnologias consolidadas nas últimas décadas (virtualização, computação em grade, cluster e processamento paralelo). Dessa forma, os provedores de nuvem pública surgiram oferecendo serviços e recursos computacionais através da internet. Ao mesmo tempo, o conceito de nuvem privada surgiu, nessa tecnologia a nuvem é implantada dentro do domínio das empresa utilizando a própria infraestrutura. Nesse estudo, as soluções para nuvens do tipo IaaS foram availiadas em relação as características e ao suporte para robustez de nuvem (flexibilidade e resiliência). Além disso, foi conduzida uma pesquisa do estado da arte para desempenho em nuvem buscando destacar os desafios e prováveis soluções para reduzir a degradação da performance de aplicações executadas na nuvem. Os resultados mostram que algumas soluções de código aberto para nuvem IaaS podem alcançar elevados níveis de robustez e podem ser adequadas para a execução de aplicações corporativas. Ainda, foi desenvolvida uma equação para encontrar o nível de confiabilidade de experimentos.

Keywords: Computação em Nuvem, Infraestrutura como Serviço, Survey, Análise de desempenho, Redes de Computadores.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

API	Application Programming Interface
AWS	Amazon Web Service
COV	Coefficient of Variation
CRM	Customer Relationship Management
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
EC2	Elastic Compute Cloud
ERP	Enterprise Resource Planning
GPL	General Public License
HPC	High-performance computing
IaaS	Infrastructure as a Service
IP	Internet Protocol
LAN	Local Area Network
LARCC	Laboratory of Advanced Researches for Cloud Computing
LVM	Logical Volume Management
LXC	Linux Containers
MAN	Metropolitan Area Network
MBW	Memory Bandwidth Benchmark
MIMD	Multiple-Instruction, Multiple-Data
MISD	Multiple-Instruction, Single-Data
NASA	National Aeronautics and Space Administration

NFS	Network File System
NVP	Network Virtualization Platform
PaaS	Platform as a Service
RHEL	Red Hat Enterprise Linux
SaaS	Software as a Service
SCM	Supply Chain Management
SETREM	Sociedade Educacional Três de Maio
SIMD	Single-Instruction, Multiple-Data
SISD	Single-Instruction, Single-Data
SMB	Server Message Block
SSH	Secure Shell
OS	Operating System
OSG	Object Storage Gateways
OVS	OpenVSwitch
QCOW	QEMU Copy On Write
QEMU	Quick Emulator
QoS	Quality of Service
iSCSI	Internet Small Computer System Interface
TCP/IP	Transmission Control Protocol/Internet Protocol
TFTP	Trivial File Transfer Protocol
VHD	Virtual Hard Disk
VM	Virtual Machine
VMDK	Virtual Machine Disk
VMFS	Virtual Machine File System
VMM	Virtual Machine Monitor
WAN	Wide Area Network

CONTENTS

INTRODUCTION

The cloud paradigm brought a new way to organize and use the hardware resources through the virtualization, which allows the provision of services on demand (BUYYA; BROBERG; GOSCINSKI, 2010). Different tools has been created in order to manage the virtual infrastructure motivated by the growing of its usage (THOME; HENTGES; GRIEBLER, 2013). Consequently, the performance evaluation is necessary because it impacts on many aspects (e.g., quality of service, best resource usage, energy efficiency, service deliverable, easier utilization, reliability, cost saving Marinescu (2013); Sosinsky (2010).

During the last years, the Laboratory of Advanced Researches for Cloud Computing (LARCC)[1] has been working on several issues related to the performance evaluation of IaaS cloud tools, including the research project High-Performance in Cloud (HiPerfCloud)[2]. The studies of Hentges and Thomé (2013) and Thome, Hentges and Griebler (2013), performed a literature review of the Open Source IaaS management tools and compared their deployment characteristics. Also, OpenStack[3] and OpenNebula[4] were installed and evaluated concerning installation characteristics such as: user interface, hypervisors types, network supports, monitoring interface, storage, and

[1]http://larcc.setrem.com.br
[2]http://hiperfcloud.setrem.com.br
[3]http://openstack.org/
[4]http://opennebula.org/

among others Hentges and Thomé (2013) and Maron, Griebler and Schepke (2014). The main contributions of them were a literature review of the IaaS tools, and a comparison of what the literature presents to what is true when deploying OpenStack and OpenNebula.

Later, the study of Maron (2014), Maron et al. (2014) and Maron et al. (2015), evaluated and compared the performance impact of OpenStack and OpenNebula cloud environments for High-performance scientific applications. The work covered: deployment, configuration, and experiments either for infrastructure isolation (network, memory, processor, and storage) as for parallel applications (distributed and shared memory programming). Moreover, in order to find significant differences between the tools, they used statistical methods to compare the performance results among the virtual instances and the native environment. Therefore, the main contributions of them were a methodology for performance evaluation of IaaS tools, and an impact analysis of OpenStack and OpenNebula on parallel and isolation workloads.

This research was also developed at LARCC, which continues previous works on this subject. The previously researches presented interesting contributions and results about cloud technologies. However, the present work is motivated by the need of a clear understanding of important research questions, which have not be discussed in the previous studies, they are the following:

- **Q1**: When the performance impact of open source IaaS tools is supposed to exist (related to the tools documentation)?
- **Q2**: Which is the robustness (flexibility and resilience) levels of the IaaS management tools?
- **Q3**: How much must be the input reliability rate for the test of significance differences when comparing the performance results?

- **Q4**: OpenStack and OpenNebula really impacts on the performance of isolated and parallel workloads such as Maron (2014) presented?

The goal of this work is to survey the open source IaaS management tools and performance state of the art for clouds. It will be able to answer the first tree research questions. Moreover, it is aimed to analyze the evaluated results of Maron (2014) with the answer of the first and third questions, so that it will be a way to answered the fourth research question. Therefore, the expected contributions are the following:

- **A survey of open source IaaS cloud tools.**

  - An extended flexibility analysis of the taxonomy proposed by Dukaric and Juric (2013) including and updating IaaS private cloud tools.

  - A method for analyze the resilience with IaaS private cloud tools comparison.

  - A method for analyze the robustness with IaaS private cloud tools comparison.

- **A survey of performance's state of the art for IaaS private cloud.**

  - Discussion of the performance overhead and variation on virtualized cloud environments.

  - Discussion of the possible solutions for minimize the performance degradation on cloud.

  - A model to give the input rate for the hypotheses test when comparing IaaS private cloud tools.

The thesis is organized in 3 chapters. The Chapter 1 presents topics about the project, such as the theme, problem, objectives and the methodology. The next (Chapter 2) presents the Literature Review, which describes the main subjects related to the theme (cloud computing, computer networks, cluster). Finally, Chapter 3 shows the results of the study, focusing on exhibit the tools comparison and the research finds.

CHAPTER1: RESEARCH PLAN

This chapter presents the methodology used for the research. The next Sections describe the details related to the study.

1.1 THEME

Robustness and performance analysis for open source IaaS cloud tools.

1.1.1 Theme Delimitation

This work explores IaaS cloud solutions looking for robustness and performance features. As a consequence, a survey of the main open source IaaS solutions was performed. Such research analyzes and compares features about the IaaS cloud tools. Moreover, there is a lack related to IaaS cloud performance and how the cloud tools can affect or not the performance. Knowing that the literature has been exploring the cloud performance, a survey (state of the art) was accomplished in order to list the issues for performance on cloud and uncover potential solutions to reduce the performance overhead.

This project was authored by Adriano Vogel as a requirement for the final undergraduate thesis report and advised by the MS.c. Dalvan Griebler in the Sociedade Educacional Três de Maio (SETREM). The period of the research was between December of 2014 and July of 2015.

1.2 PROBLEM

Currently, is attractive for corporations take advantage cloud computing technologies. Mostly by the combination of small financial investments with the computational resources provision. Provided by cloud delivers, this technology offers accessible infrastructure, platform and software services such a way that clients only will pay-per-use. Also, it avoids large investments in hardware infrastructure and deployment time.

However, the performance of clouds has become a challenge due to its additional abstraction layers. The first is the virtualization, which abstracts the hardware layer so that the OS will see the infrastructure as hardware components. In fact, this layer is above the hardware and may impact in the overall performance compared to the native system environment. Additionally, the performance impact and variations induced by cloud tools affects the overall system results (response time, hardware utilization, QoS, energy consumption, among others). As a consequence, the cloud performance concerns are still increasing and must be carefully analyzed.

This research was performed at LARCC. The previous studies developed on this lab explored interesting aspects of the IaaS clouds. Nevertheless, there are several subjects unclear that needs to be analyzed. As a consequence, this paper aimed to explore and address open issues.

Considering the increasing interest for efficiency on resource utilization and performance, this study intends to seek for information about new tools, explore different features, and present a analysis for performance on IaaS cloud. Also, a survey was conducted and analyzed the IaaS cloud solutions components and support for robustness. Additionally, this paper presents the state of the art of open issues (eg., overhead, degradation) for performance on cloud.Thus, the main problem can be abridged as: which are the performance and the support for robustness levels of the IaaS cloud tools?

1.3 HYPOTHESES

- The open source private cloud solutions for IaaS will not impact on the performance when running workloads, according to their description.
- All the surveyed cloud tools offers the same robustness levels.
- 95% of reliability is the appropriate indicator for hypotheses test to compare significant differences between open source IaaS private cloud solutions.
- OpenStack and OpenNebula impact on workloads' performance as presented by Maron (2014).

1.4 VARIABLES

- Performance
- Robustness
- Reliability

1.5 OBJECTIVES

This Section describes the study goals, where they are presented as items following the chronological order, and as they are expected to be achieved.

1.5.1 General Objective

The goal is to survey open source IaaS cloud tools and analyze the support for robustness on the tools, and the performance impact of the private cloud IaaS tools.

1.5.2 Specific Objectives

- Study cloud computing and performance measurement.
- Study open source IaaS tools.
- Survey open source IaaS tools.
- Study related work.
- Survey the robustness of IaaS open source cloud tool.
- Analyze the cloud IaaS solutions for potential performance impact.
- Understand what actually the IaaS tools handle and how they interact with the layers below (virtualization, hardware).
- Determine the factors of the performance overhead on the cloud.
- List possible solutions to deal with the performance degradation on virtual environments.
- Document all steps of the research.
- Write papers.

1.6 JUSTIFICATION

As emphasized by Buyya, Vecchiola and Selvi (2013): "Cloud computing is the most recent emerging paradigm promising to turn the vision of “computing utilities” into a reality". Today, companies depend on powerful and efficient computational resources for processing huge amounts of data (big data) or delivering services. Using cloud computing technologies, their resources available may increase resulting on business, new products and money saving. Furthermore, when moving their infrastructure to a cloud environment or deploying a cloud, they can achieve the following advantages:

on-demand self-service, multiple devices and platforms support, dynamic and flexible resources allocation, fast elasticity and deployment, energy saving, and among others (SOSINSKY, 2010).

As cloud computing paradigm started the service delivering process (IaaS, PaaS or SaaS), the Quality-of-Service (QoS) became a prerequisite of both providers and users. One of the most important aspects of QoS is the performance, which may impact on all previous advantages promised by the cloud adoption. Also, the robustness became an important aspect in order to address the QoS required for the tenants . In addition, the stack of service models indicates that IaaS has weight on the two upper layers because they work on top of its implementation.

Regarding to data center aspects, hardware and virtualization layers may also impact on the performance, but their impact has been evaluated during recent years by many researches (Xavier et al. (2013); Regola and Ducon (2010); Kovari and Dukan (2012)). However, the performance evaluation of IaaS management cloud tools continue under research. Mostly because its measurement is complex and requires a careful analysis over the environment characteristics bias, for example, hardware configuration, network implementation, storage model, and virtualization technology. Moreover, the cloud still as a recent and emerging paradigm, which means that the tools and the overall system may have inconsistent or performance variation.

Currently, there are trends for increasing the computational power, the energy efficiency and reducing the hardware investments (costs). In order to achieve high performance, the system can not consider only for hardware and software settings. New paradigms have been emerging and they need to be considered when building a data center (operation, efficiency, energy consumption, communication, resource management, user, virtualization control).

The cloud computing paradigm has become an alternative to achieve computational high performance and efficiency.The virtual pool powered by IaaS tools may run either HPC and enterprise applications. Currently, it is evident that cloud is suitable for run enterprise workload. On the other hand, the cloud for HPC workloads is still under research and the challenges to overcome are many (performance variation, degradation, overhead) because of the additional abstraction layers (virtualization, management, scheduler, APIs, security, and among others). For this reason, the demand for IaaS cloud analysis by the scientific and enterprise field is increasing and it will still growing during the next years (FURHT; ESCALANTE, 2010). Also, each cloud solutions works and it is deployed on a different way (the technologies impact on performance and the performance affects the pursued efficiency). Such variations affect cloud users and IT managers in the decision-making process of what deployment is the most suitable.

This open source infrastructure management cloud tools and information related to free solutions. At first, the options available need to be analyzed concerning features and the support for robustness (surveying). Furthermore, a survey for performance was performed in order to highlight the main challenges for performance on cloud and emphasize possible solution for increase the performance and measure the reliability rates.

1.7 METHODOLOGY

This Section describes the main methods used to achieve the goals. In addition, it will explain the research procedures to carry out hypotheses and perform the proposed research (study, install, tests, comparison, conclusion).

1.7.1 Methods

In this research, it was used the quantitative method because the main goal is to look for QoS on cloud systems. Such subject is difficult to measure due to the fact of the several architecture layers present on a cloud system. In addition, it was used the deductive method to analysis the finds and evaluate the IaaS cloud management tools from the respective documentation.

The validation of the hypothesis shares some characteristics. For example, the environment (hardware components, network, host OS) needs to be equivalent. Also, the research questions and problem are complex. In order to address such challenges, the Hypotheses and the way to answer are presented below:

- The open source private cloud solutions for IaaS will not impact on the performance when running workloads, according to their description.

  This hypothesis is validated over the analysis of the tools documentation.

- All the surveyed cloud tools offers the same robustness levels.

  In order to validate this hypothesis, the IaaS cloud tools features need to be analyzed concerning their support for robustness (flexibility and resilience) and rank the differences among such tools.

- 95% of reliability is the appropriate indicator for hypotheses test to compare significant differences between open source IaaS private cloud solutions.

  This hypothesis might be validated over a performance analysis of workloads performance impact on cloud and native environment.

- OpenStack and OpenNebula impact on workloads' performance as presented by Maron (2014).

  Such hypothesis may be solved over an evaluation of the performance reliability rates for cloud and apply this method on Maron (2014) experiments results. Also, this hypothesis can be validate over a equation which provides the maximum reliability rate and compare between the different measurement environments.

1.7.2 Procedures

Exploratory research: This procedure was used because of the need for exploring either the cloud solutions features and also analyze the aspects related to the performance degradation.

Literature review: There are several solutions for cloud and virtualization issues on the literature. Several researches have been performed in order to solve specific challenges. As a consequence, in this thesis it was explored such papers and highlighted the solutions available.

1.7.3 Research Techniques

To validate the hypotheses, it was used the technique of experiments because it is needed to measure the performance impact and calculate if there are degradation or variation. Also, a technique used was the exploration of the cloud tools features and components.

1.8 RESOURCES

During the study, it was necessary to achieve the goals and, to do that, it needed some resources which are described below:

1.8.1 Human Resources

The human resources are the advisor, professors and university staff.

1.8.2 Material Resources

The main material resources are computers, books, sheets, papers and IT components.

1.8.3 Institutional resources

The institutional resources are the library, the Internet, IT labs and coordination of the Computer Networks course.

1.9 SCHEDULE

The Table 1.1 shows the schedule of activities for this study. Following, the provided time will be possible to achieve the goals. The painted cells show the expected period for each activity, and the character ”x” represents the objectives already reached. Also, the “W” represents the paper writing as the “P” means the presentation of papers.

Activity	2014		2015							
	Nov	**Dec**	**Jan**	**Feb**	**Mar**	**Apr**	**May**	**Jun**	**Jul**	**Ago**
Write a research project	x	x								
Deliver the research project		x								
Study open source IaaS cloud tools		x	x	x						
Study of related works.		x	x	x						
Present and deliver the research progress.					x					
Survey the IaaS solutions						x	x	x	x	
Survey the performance challenges for workloads on cloud							x	x	x	
Analysis and validation of the hypotheses.									x	
Write the final thesis report	x	x	x	x	x	x	x	x	x	x
Write and Publish papers	x(W)	x(W)				x(P)	x(W)	x(P)	x(W)	
Present the final thesis report										x

Table 1.1: Activities schedule.

1.10 ESTIMATED COSTS

The estimated financial resource needed to accomplish the project is showed below on Table 1.2.

Activity	**Amount**	**Unit value**	**Total value**
Number of printouts	600	R$ 0.15	R$ 90.00
Spiral Binding	5	R$ 4.00	R$ 20.00
Hardcover binding	2	R$ 70.00	R$ 140.00
writing and publish articles	3	R$ 400.00	R$ 1.200.00
Work hours	450	R$ 45.00	R$ 20.250.00
Total	R$ 20,950.00		

Table 1.2: Estimated costs.

CHAPTER2: LITERATURE REVIEW

2.1 DEFINITION OF TERMS

This Section defines the terms which are related to the research. It explains the main characteristics and details of the topics.

2.1.1 Operating System

According to Hansen (1973), the Operating System (OS) is a set of interactions and commands that enables the user to control the hardware resources through an interface. Also, OS allows people to share computational resources, where users will be obliged to follow a policy of resources utilization and privileges to request the processor, memory, network or storage. This policy sometimes locks specific process, or files from specific users to avoid conflicts in programs and files. Currently, the operating systems can handle multi-user and multi-tasks in order to achieve the best hardware utilization.

2.1.2 Computer Networks

According to White (2012), a computer network can be defined as a connection of devices and equipment through wires or radio waves, with the main goals of transfer data and share resources. Networks can be built using different topologies (e.g. Star, Tree, Point-to-point, among others), and different layouts (e.g. LAN, WAN, MAN). Also, a network architecture uses a standard to organize its functions and services. Firstly, it was created the Open Systems Interconnection (OSI) model that divided the structure in 7 layers. And after that, it was developed the TCP/IP model of protocols based on OSI, which is used for Internet connections around the world. Both models are described as follows:

- **OSI Model**: According to Dean (2012), the OSI (Open Systems Interconnection) was created to help to understand and developing computer-to-computer communications over the network. It is divided into seven layers: Physical, Data Link, Network, Transport, Session, Presentation, and Application. Each layer has protocols that perform services. The protocols interact with protocols on the layers directly above and below. The OSI model represents what happens between two nodes communicating on a network.

- **TCP/IP**: Kozierok (2005) describes TCP/IP as a stack of different protocols, where two are usually considered the most important (TCP and IP). The Internet Protocol(IP) refers to the third layer of the OSI model, which is the network layer protocol. It is responsible to provide the address, datagram routing, and other functions on the Internet world. The Transmission Control Protocol (TCP) is the main transport protocol layer, which is the layer 4 of the OSI model. It realizes the connection establishment (data transfer) and management (congestion control, flow control), also the checking data process (error detection) between software and devices.

2.1.3 Performance Evaluation

The performance of computational systems can be determined by the capacity of process requests in less time (efficiency). Also, the performance efficiency impacts directly in the corporation production. So, if the infrastructure, virtualization technology, OS and applications are properly set the resources utilization will be better.

Furthermore, the softwares provide resources and tools, which is often under direct interaction with the user. If the systems are well-optimized, the users will have the best capacity and experience (LIU, 2011). Also, the main evaluation parts of a whole system are described below.

2.1.3.1 Response Time (Latency)

An important metric for computer systems is time. It occurs because time is a key related to performance and QoS (Knowing the many external variables which can affect the results). The conceptual definition of response time is the elapsed time between the program is started and when it is answered or returns what is expected.

Additionally, the response time is often measured by the average of repeated operations using a random period of time (FORTIER; MICHEL, 2003). The response time is related to the performance of a Latency of a system. Also, the "N" refers o what is transfered in the time period.

$$latency = \frac{N}{time}. \tag{2.1}$$

The throughput and latency are pretty important for the completion time, because of the dependency among such factors. The completion time is the while taken between the start and end of any job. Also, when considering huge jobs the completion time may be over-estimated Haring and Kotsis (1994). Moreover, the queue scheduler manager may impact on the completion time due to the variance on the resource allocation.

2.1.3.2 Throughput

Throughput is defined as the maximum transfer rate achieved transferring data between different locations on a defined period of time. In other words, throughput or bandwidth is a measurement of the estimated performance, related to how many jobs might be done in predefined time period (PATTERSON; HENNESSY, 2013) . It is commonly used to measure the performance of the computational system and components (memory, hard drive, network). Since the last decades, the pressure for higher throughput has increased. As emphasized by the authors Olukotun et al. (2007) became a challenge to achieve the expected throughput and different techniques have been emerged to fill the lack (multi-cores processors, higher memory frequency, SSD storage, among others). The throughput can be find by the division of the latency for Ni (the size of the data)

$$Throughput = \frac{latency}{Ni(bits)}. \tag{2.2}$$

2.1.3.3 Speedup

According to Lilja (2005) speedup is a metric for test the computational performance among systems. This measurement consider throughput or speed on the execution time. Moreover, the main goal of this metric is to find the faster computer or distributed combination in order to increase the applications results.

In addition to that, the speedup is defined by the Amdahl's law as a gain of performance when executing tasks on heterogeneous environments (Hennessy, Patterson and Asanović (2012)). Also, the speedup equation is presented below:

$$Speedup = \frac{Performance(improved)}{Performance(normal)}. \tag{2.3}$$

2.1.3.4 Efficiency

As emphasized by Fortier and Michel (2003), efficiency is another important aspect of computational performance measurement. Such metric is based on the resources utilization and the higher throughput rates achieved.

$$Efficiency = \frac{throughput(real)}{throughput(theoretical)}. \quad (2.4)$$

In addition to that, there is available another equation in order to get the efficiency on parallelized system. This equation unless of consider the throughput, takes into account the speedup rates (MCCOOL; ROBISON; REINDERS, 2012).

2.1.4 Resource Isolation

Since the emergence of computational services the programs, applications and services still demanding an increasing computational power. Such demand is a challenge for IT professionals, CEOs and also scientists mainly because the enterprises depends on computers to run their tasks.

As a trend for virtualization became almost impossible to achieve physical computer isolation. It happens because of the high costs and low utilization of dedicated hardware (Raj et al. (2009)).

As a consequence, has been emerged the virtualized resources isolation which has been largely used. Despite the fact of not having the same results as a physical isolation, the virtual isolation is a trend driven by the advantages that virtualization brings (ease deployment, flexibility, live migration, fault tolerance, among others).

Finally, the isolation can be defined using ideas given by the author Chevance

(2004). Such definition presents that when a data transaction occurs, to change or access the values is only possible when the process has been finished (unavailable to other transactions before the first ends.)

2.1.5 Statistical Significance

Statistical significance is often confused with scientific discovers. However, it is not the same thing (ZILIAK; MCCLOSKEY, 2008). Also statistical significance is defined as the most common quantitative way to analyze if data looking for the distinction between samples. The statistical significance testing in the begin assumes that no differences exists at all and tries over the analysis find contrasts.

Additionally, the hypotheses statistical significance tests are indispensable and assumes that the hypothesis is false. Such assumption gives the idea of null hypothesis (no differences). Also, the results of the experiments answer the research problem, when the such results are not the expected, the researchers can reject the null hypothesis and assume that exist differences in the comparison (RIEGELMAN, 2005). In other words, the author Field, Miles and Field (2012) explains the fact of use experiments to validate the hypotheses through a statistical analysis. Such statistical analysis may confirm or not the validation of a whole theory.

2.1.6 Survey

Survey is often characterized as a research to collect information. Also, it can be a discussion or a conversation related to a specific theme. The surveys methods may be applied to a large population or specific topics and it requires a research goal. Also, the surveys' answers can be seen as an output or feedback, depending on the specifics characteristics of the survey. Thus, the survey can be used as reference for a research to facilitate the decision-making because and it presents results that brings information about interesting subjects (FOWLER, 2008).

2.1.7 Open Source

WEBER (2009) presents in his book a citation related to code openness: "Proprietary software ran directly against the moral sentiments of a decent society". In addition of this, the author presents the concept of free software, which means freedom to run, distribute, and modify everything. However, this freedom does not necessary means gratis (no cost) because the developments may sometimes need to sell open source software to grab financial resources to still coding for new solutions.

Currently, one of the most popular free licenses of software is the GPL type: "Software that is licensed under the GPL cannot be made proprietary" (WEBER, 2009).

2.1.8 Quality-of-Services

A definition given by the author Parsons and Oja (2008) explains QoS as the performance results that a provider offers to customers, which impacts directly on the programs execution. Additionally, the key challenges for a good QoS levels are reliability (allow the customers to trust the system) , availability (stay working for people use) and serviceability (easy and fast maintenance).

As defined by Sosinsky (2010) "The Quality of Service (QoS) is something that you can obtain under contract from your vendor". The idea of QoS is a constant process of improving the services. Such improvements will allow the company or a service provider to reach a reference due to its efficiency.

If someone aims to achieve a real Quality of Service on cloud, the first step is to think in the future scalability. It is important because the majority of IT workers only have financial resources to maintain the services and resources available. However, the IT division inside the companies is much more than that. The IT has to be seen as strategic businesses operations. The demand for resources is still increasing the investments. The Quality of Services relies in an efficient plan and correct infrastructure utilization (BUCHTA; EUL; SCHULTE-CROONENBERG, 2010).

In the real world, the need for QoS is still growing fast. Another aspect of such increase is the IT services and technologies changes. One of the biggest milestone is cloud computing, which offers infrastructure and programs over the internet. Also the cloud may be suitable for high performance application. However, the cloud demand for QoS is high and the complexities to build and maintain such systems working properly is pretty hard due to fact of the resources sharing and multi-tier clients and applications. This idea is presented by the authors of Liu, Chen and Yang (2012), where is a challenge to keep the performance needs for scientific workloads running on heterogeneous cloud.

On the other hand, if the public providers will still facing challenges related to QoS, for customers who need high QoS levels, a private cloud could be a solution (ABRAHAM; MAURI; BUFORD; SUZUKI; THAMPI, 2011). It is because in such systems the cloud is deployed inside the corporations domain. As a consequence, if the servers are powerful and the software layers are configured properly, it can be easier to guarantee performance.

2.2 BACKGROUND

The demand for computing power continues to increase since the first computer was built. Currently, the scientists from different fields of study share the same need: computational resources. The challenge of IT professionals is to supply this demand. Moreover, it is necessary the development of others technologies, such as clusters, computing grids and cloud computing. These technologies have emerged in the last decades to increase efficiency and computational performance Vicat-Blanc et al. (2013). To understand the topics of this work, its necessary, initially, to conceptualize some subjects, as presented in the next paragraphs.

According Hansen (1973) Operating System is a set of procedures which makes possible the resources of a computer be shared between a group of people efficiently. Despite that the users are competing for computational resources as processor and storage space, the operating system makes possible to use the same database or the same program for more than one user, for example. However, the operating system must need to have a policy to determinate the order to execute the tasks, been responsible for solving problems involving simultaneous requests of the resources.

On the other hand, it is necessary to understand the reason and the objective of another topic that will be cited in this paper: the parallel computing. Using the nature as the model, where with the division of the tasks is faster and easier to finish them, the computer area implemented the same concept in the processing: it would be much easier and faster to finish the tasks if they are divided in small pieces and each one of this pieces being solved concurrently by different processors (MARINESCU, 2013).

2.2.1 Computer Architecture

The author East (2004) wrote a whole book related to computer architecture. Among several chapter and topics, we can abstract a lot of important information for this work. The book defines a system as a component responsible that "generates output from the input". In other words, each system input value will soon or later (depending on the response time) return an output to users or another process (program). over an equation The overall system performance has a limitation related to vertical growing (increase the processing power). However, since the computer emerged, the demand is still increasing for computing power and high performance. As consequence, the number of processor units (multiprocessor) has increased (CHEVANCE, 2004), in order to improve the performance.

In addition, Chevance (2004) presents that one architecture of multiprocessor as Tightly-coupled, or Symmetrical Multiprocessor (SMP). Thereby, all the cores are allowed to see and explore the shared system resources(memory, I/O). Also, this model is based in just one Operating System which manages all the infrastructure and resources. It is a cost-effective (economic) solution to improve the performance.

Another is the Loosely-coupled which is based on separate hosts interconnected over a high-speed network. Such hosts are characterized as independent systems in such way that each one has its own resources and OS. Also, the users have the transparency as if only one system was running.

A computational architecture is designed by a set of components that work together as a unit to solve the related challenges.

- **Processing Unit**

  The authors Abd-El-Barr and El-Rewini (2005) emphasize that a typical CPU is shaped by three main parts:

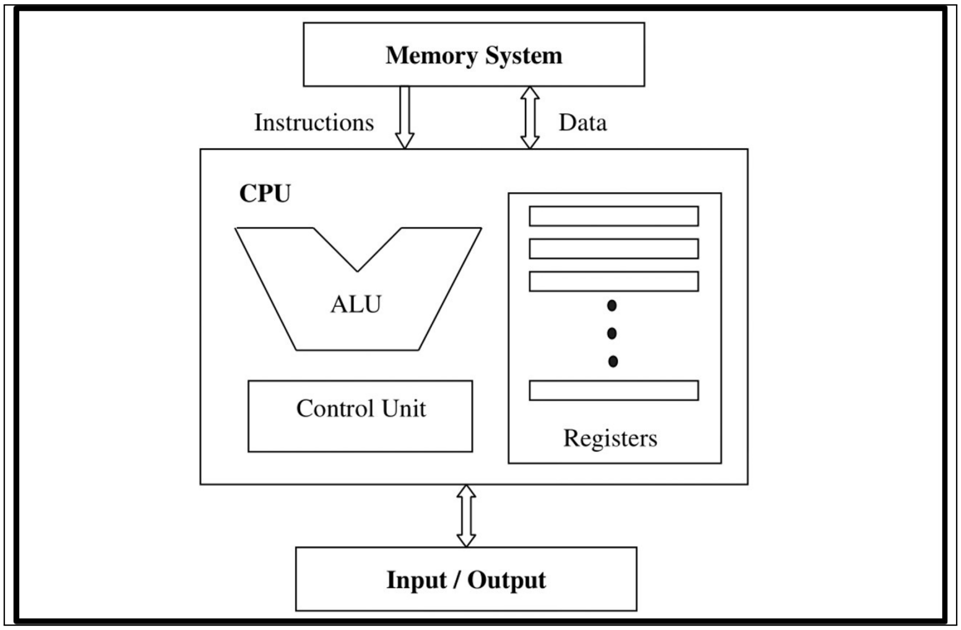
  - Register set: Provides the channel to store and hold instructions of processes that are running or paused

  - Arithmetic logic: is compound by the circuits which perform the processing and instruction tasks (arithmetic, logical).

  - Control unit: is a part which executes and decodes the instruction to main memory.

  In other words, processor units are the main part of a computer system. Such units are responsible to receive the input and transform it in an output (results), as math calculations, float points, data access, among others.

  During the last decades, the servers processing power has doubled due to hardware evolution (Moore's Law). Moreover, the overall system memory capacity and bandwidth has improved (CHEVANCE, 2004). Such improvements of the computational power are directly related to the demand of the market. Also, the processor power had a faster improvement compared to memory resulting



Source: Abd-El-Barr and El-Rewini (2005, p. 84)

Figure 2.1: CPU components and interactions.

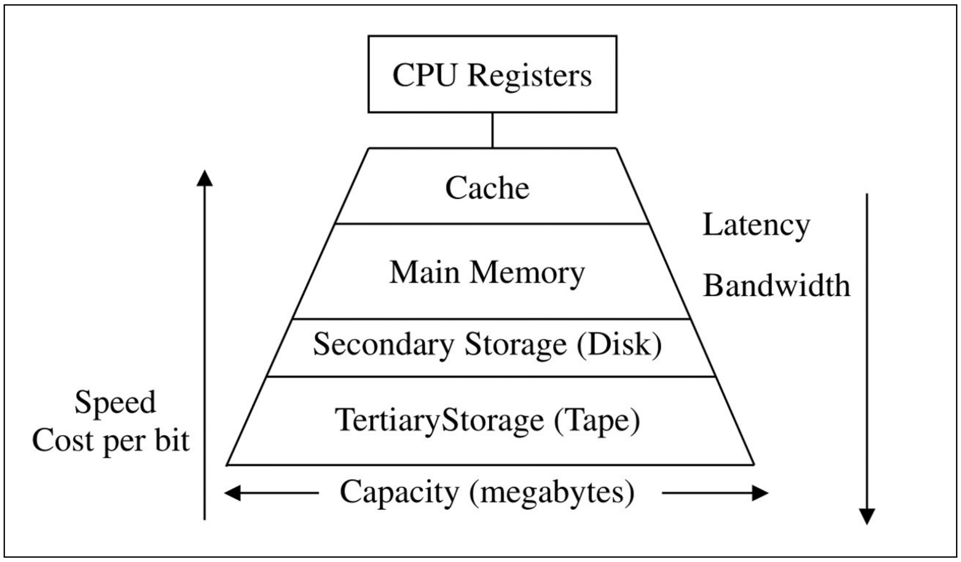
in a performance lack. As a consequence, the computing architectures include a cache technology. Such solution, saves the most used processing results in a fast and easy place, often inside the CPUs.

- **Memory System**

As described by the authors Abd-El-Barr and El-Rewini (2005), the memory can be explained as a hierarchy: "The memory hierarchy can be characterized by a number of parameters. Among these parameters are the access type, capacity, cycle time, latency, bandwidth, and cost".

The first part is the cache memory which is fast and small. The other part of the hierarchy is the main, characterized as larger and slower. Both are built using semiconductor material which can defined as the fast memory type named primary memory.

Finally, another model of memory is the stored on disks (secondary memory) or tapes (tertiary memory). This type is cheaper, larger and slow compared to the solid-state model.



Source: Abd-El-Barr and El-Rewini (2005, p. 108)

Figure 2.2: Memory hierarchy.

- **Input – Output (I/O)**

  The I/O according to Chevance (2004) are one of the most important components of a computer system. It is because of the dependency of such devices and lack of researches in this field.

  Due to the I/O typical system comprises several machine physical components. The main are described below: - System controller: shaped by chips which integrates and connects the different controllers (processor, memory and I/O bus). - I/O controllers: this type is used to connect internal buses (PCI, Network adapters, among others peripherals).

- **Data Storage**

  The act of store data on the computer system is a part related to I/O. Also, the data became one of the most important aspects of corporations. It pulls the levels of challenges for keeping the data available and secured.

  Currently, there are several different storage systems, the main used for servers are NAS (Network-Attached Storage), SAN (Storage Area Network) and iSCSI (remote disks over the internet).

Also, the authors Chevance (2004) introduces the concept of Storage Virtualization. Such technology uses the virtualization advantages (Flexibility, elasticity, scalability, fault tolerance) to store data among blocks.

According to the author Sabharwal (2013), the storage in cloud technology based on solutions which are often performed using the cluster storage paradigm which may combine several categories, such as local storage (in the slave host) distributed or shared storage over the network. Also, several vendors started to offer cloud storage services. In such services, the client pays a billing according the storage capability which has been selected.

2.2.2 Flynn's Taxonomy

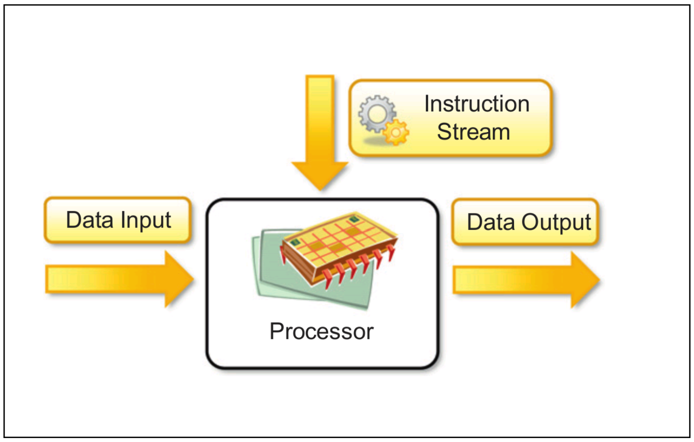
According to Mattson, Sanders and Massingill (2004), the Flynn's taxonomy is the most classical way to characterize the parallel computing architectures (supercomputers, clusters, workstations, distributed architectures). As defined the same authors, the Flynn's taxonomy: "He categorizes all computers according to the number of instruction streams and data streams they have, where a stream is a sequence of instructions or data on which a computer operates". Among them, there are available, SISD, SIMD, MISD, and MIMD which are described below.

- **SISD - Single Instruction, Single Data**

  This model is commonly used on single-processor architectures. Such system is based in the concept that each stream of instruction is in charge of only one stream of data each time.

- **SIMD - Single Instruction, Multiple Data**

  This system uses a type of parallelism through interprocess communication. Also, each instruction stream is broadcasted to the multiple parallel processor which are managed by a single control unit.



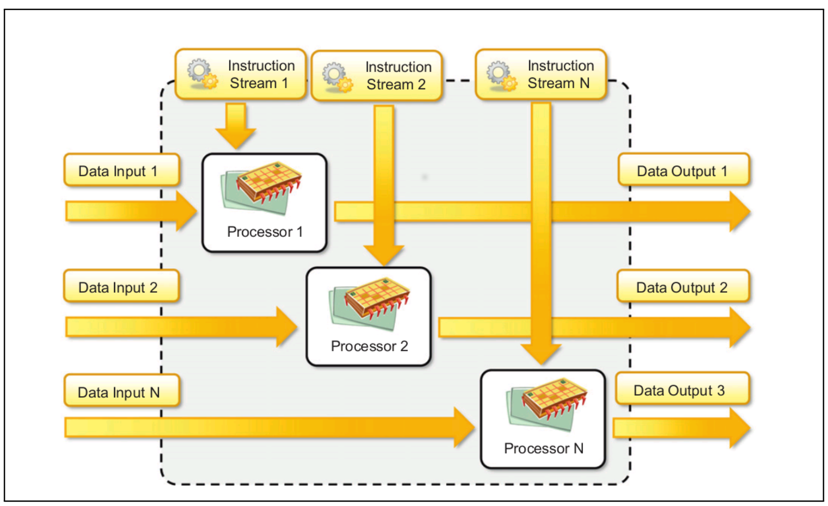
Source: Buyya, Vecchiola and Selvi (2013, p. 32)

Figure 2.3: SISD architecture.

- **MISD - Multiple-instruction, single-data**

  In this model, as emphasized by Buyya, Vecchiola and Selvi (2013), the machines are not used for business or home users because they are useless. Such fact happens because this model "became more of an intellectual exercise than a practical configuration" (BUYYA; VECCHIOLA; SELVI, 2013).

- **MIMD - Multiple Instruction, Multiple Data**



Source: Buyya, Vecchiola and Selvi (2013, p. 35)

Figure 2.4: MIMD architecture.

This architecture is used in most parallel systems which are currently deployed.

In this system, each processor unit as its control unit which sends the instructions.

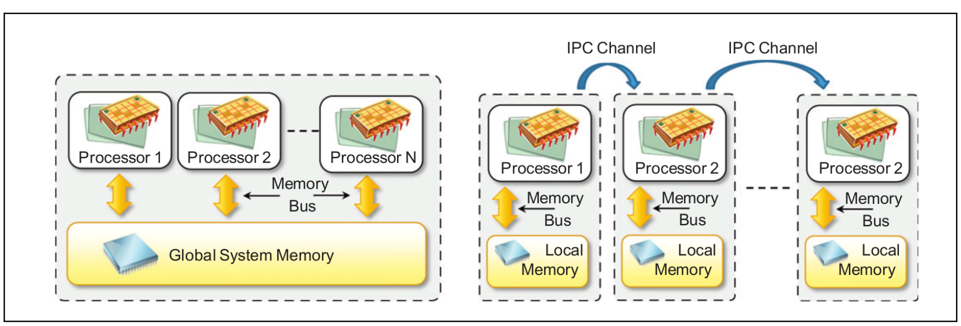
This category fits inside different models segregated by the memory organization used.

One category is the shared memory, which according to Mattson, Sanders and Massingill (2004) is: "all processes share a single address space and communicate with each other by writing and reading shared variables".

SMP (symmetric multiprocessors), is the whole number of processors shares a link to the shared memory using the same channel speed. It is a simple class resulting in the fact that it does not scale memory properly and only supports a reduced number of processors.

Another category of shared-memory is the NUMA (nonuniform memory access). Such class of shared memory is randomly accessed and modified from both processors which brings advantages related to the throughput rates and support of a large number of CPUs. On the other hand, the response time between the processor and memory is unpredictable, since the location of the physical memory block varies from the processor, it impacts directly in the response time .

Finally, according Mattson, Sanders and Massingill (2004) in the distributed memory system the process is allowed to access and lock a memory block in such way that it belongs to him. Also, the different processes communicate between them sending and getting messages.



Source: Buyya, Vecchiola and Selvi (2013, p. 35)

Figure 2.5: Shared (left) and distributed (right) memory MIMD.

2.2.3 Computing Applications

According to the authors Hoggatt and Shank (2009), computer applications or software means exactly the same subject. It is a program which executes jobs for a client or to another program in a desktop environment (eg., text editor, internet software, database, games, among others).

The applications run and are controlled by an operating system. The OS manages the software in such way that defines how many resources they may use and the files permissions (read, write, execute).

Looking to the cloud paradigm, the computer applications are made available to users in a revolutionary way. Such innovation, brings new concepts, services and challenges to providers. On the other hand, it brings to users a wide and variety of options and applications to choose. When using cloud for resources provision, the applications (programs) are offer as services (eg., infrastructure, platform, software), more details are available on Section 2.2.5.

2.2.3.1 Scientific Parallel Applications

A defined by the authors Heroux, Raghavan and Simon (2006), Scientific computing is: "a broad discipline focused on using computers as tools for scientific discovery". Since the scientific computing emerged, several ways and applications arose to enrich this concept. Among several options, there are the parallel applications which are algorithms and tools aiming for offer functionality that increases the overall performance of the systems (eg., portability, interoperability, flexibility).

In addition, the parallel scientific applications are used in modern computational approaches (clusters, supercomputers, grid environments) to increase the performance of the tools and increase the Quality of Services.

Furthermore, concerning parallelism, the author Bisseling (2004) defines as: "the use of more than one processor to solve a problem", it is directly dependent of an efficient programming model. In other words, the people dealing with the development of scientific parallel applications should be aware of the hardware resources (processor time, memory).

Nevertheless, the parallel processing is still a challenge, it is not that simple as sequential programming. Despite the difficulties, several researchers continue working in ways to improve the programming model, and the programming libraries and languages. As a consequence, some points have been realized, such the efficiency, plainness and the integration with all the parts need to be combined (hardware engineers, developers, users) (BISSELING, 2004).

Concerning to parallel processor designs, the authors Reed and Fujimoto (2004) give three different types related to the number of CPUs and complexity related. The first and simplest is the bit-serial unit processor. Contrasting with the simplicity, the computer power achieved may be massive by the fusion of many units.

On the other hand, there is available a way that uses a reduced numbers of processors. However, each processor handles a huge amount of data and contains an efficient pipelined architecture. The last is the most common approach. This way of parallel programming binds a bigger number of microprocessors in a distributed multicomputer network.

2.2.3.2 Enterprise Applications

Over the last decades, the enterprise field needed to rely on IT to offer and process its applications and softwares (eg., ERP, SCM, CRM, text editors, Internet browsers, files storage). In addition, currently there is a trend to run the enterprise application and operational tasks in the cloud platforms (SHROFF, 2010).

In short, enterprise applications are the top strategic business processes inside a corporation and they are presenting a fast growing. Also, the applications processes surely need to communicate and integrate other business applications of the same company. For this reason, to achieve the necessary integration it is important that companies explore either the concept of an application running on a cloud or on its own data center.

Also, as emphasized by the huge cloud handbook Furht and Escalante (2010), either when running scientific and enterprise application, it produces often a large amount of data-sets. The information beyond such workloads is a vital source of market knowledge and scientific consideration. However, in order to have this results several steps are needed. One of the main is the data analysis (data mining) and using such analysis results offer on demand solution (big data).

2.2.4 Benchmarks

The authors of Kanoun and Spainhower (2008), provided a sharp explanation related to computers performance evaluation using benchmarks. Such benchmarks are motivated by the current dependability of computer applications and services. Also, benchmarks are considered by the computer industry a reliable way to measure and compare systems. The results of benchmarks are expressed on a quantitative or qualitative way. Such results aim to test the system, for example, to find an eventual weakness, fault tolerance, applications behaviors, among others.

2.2.4.1 Resource Isolation

As emphasized by Sakr and Gaber (2014) the resource isolation is managed by the OS to efficiently control and fair distribute the tasks and system calls (allocation, scheduler).

The STREAM benchmark is a "Real World" benchmark which evaluates the memory bandwidth (MB/s) using vector kernels. (MCCALPIN, 1996) It is possible to be performed using larger data sets than the cache system, in such way that forces the increasing of the memory usage.

The Iperf (2014) is a benchmark for measure TCP and UDP bandwidth, latency, speed, and datagram loss. Also, Iperf promises to increase the accuracy of the results, using a more efficient way to evaluate the performance of the network.

The benchmark Iozone (2006) measures the Disks throughput. It covers all file system, since physical blocks to virtual volumes or datastores. IOzone is supported by many architectures and operating systems.

The Linpack benchmark simulates the processing of large matrices and linear equations (floating points). It uses techniques of stress the CPUs in such way that gets much processor speed as possible. (DONGARRA; LUSZCZEK; PETITET, 2001)

2.2.4.2 Parallel and Enterprise Benchmark Suites/Workloads

As emphasized by Culler, Singh and Gupta (1999) there was a time defined by the difficulty of find a reliable parallel software. Fortunately, after a while several types of parallel benchmarks has been launched (multiprocessor, shared memory, distributed memory).

- **ScaLapack**

  This suite is based on a deployment which simulates linear algebra operations (equations, problems, matrix factorization) on machines kernel Culler, Singh and Gupta (1999). Also, it is/was distributed by Oak Ridge National Laboratories.

- **TPC**

According to the same authors, the Transaction Processing Performance Council (TPC) is a combination of a bunch of different benchmarks classes (TPC-A, TPC-B and TPC-C). These benchmarks simulates a wide type of workloads, such as a database (TPC-VMC) and system processing (NAMBIAR; POESS, 2013).

- **SPLASH**

  As Culler, Singh and Gupta (1999) wrote, the SPLASH (Stanford ParalleL Applications for SHared Memory) is a suite of benchmarks developed by the Stanford University focused on cache test. Also, it is written in C and exploring the parallelism of systems.

- **NAS Parallel Benchmarks**

  This benchmark suite was developed by the NASA (National Aeronautic and Space Administration). As defined by Culler, Singh and Gupta (1999): "They are a set of eight computations, five kernel and three pseudo-applications".

  The kernels simulate problems to solve such as equations, three-dimensional, integers number, and the data set may be customized according to the goals. The common usage of this suite is to compare machines overall performance.

- **Phoronix Test Suite**

  In this approach, an entire suite of benchmark has been bound together in order to offer a complete test option Russell and Cohn (2012). In addition, the Phoronix Test Suite is a framework developed over a flexible architecture and is highly customizable, focusing on Linux performance and supporting several others platforms.

  Furthermore, the author Kukunas (2015) emphasizes this suite advantages (straightforward benchmark options) and importance (compile and present graphs).

- **Parsec**

  The Princeton Application Repository for Shared-Memory Computers (PARSEC) is a bundle of tests to analysis Chip-Multiprocessor available for public in order to pursue the high performance Bienia et al. (2008).

Also, Parsec is a benchmark suite shaped by five multithreaded workloads for shared-memory and multiprocessors (HERKERSDORF; RÖMER; BRINKSCHULTE, 2012).

- **Yahoo! Cloud Serving Benchmark (YCSB)**

  This benchmark is focused on simulate the support for run high demands workloads (database, data mining, XAP) YSCB (2015). The main goal of YCSB approach is to provide a framework and emulator of synthetic tasks for performance analysis of application running on a cloud system. The most common way is use one or more clients that execute the system requestion and save a log of the results. In the end, several results may be compared in such way that the average and best scenario is found.

2.2.5 Cloud Computing

The word cloud was used to represent the abstraction of the network in the telecommunications. After that, it was recognized as the symbol of the Internet, that is fundamental in cloud computing because it is the environment that allows to deliver many services to the users. Cloud computing becomes a referential to the applications and the system involved in the process to delivered the services to the users. By this concept, it is possible to measure the services and priced them, using a strategy “pay-per-use”, where the user access an online storage or development platforms and pay only for the effective usage (BUYYA; VECCHIOLA; SELVI, 2013)

The author Sosinsky (2010) defined Cloud Computing as applications that are not physical accessible to the user. In other words, the user can run applications, can make use of the resources that are available. However, the users do not have access to the physical equipment. The resources offered may be allocated on servers around the world. Such applications, using Internet protocols, may be accessed and run on a distributed virtual network.

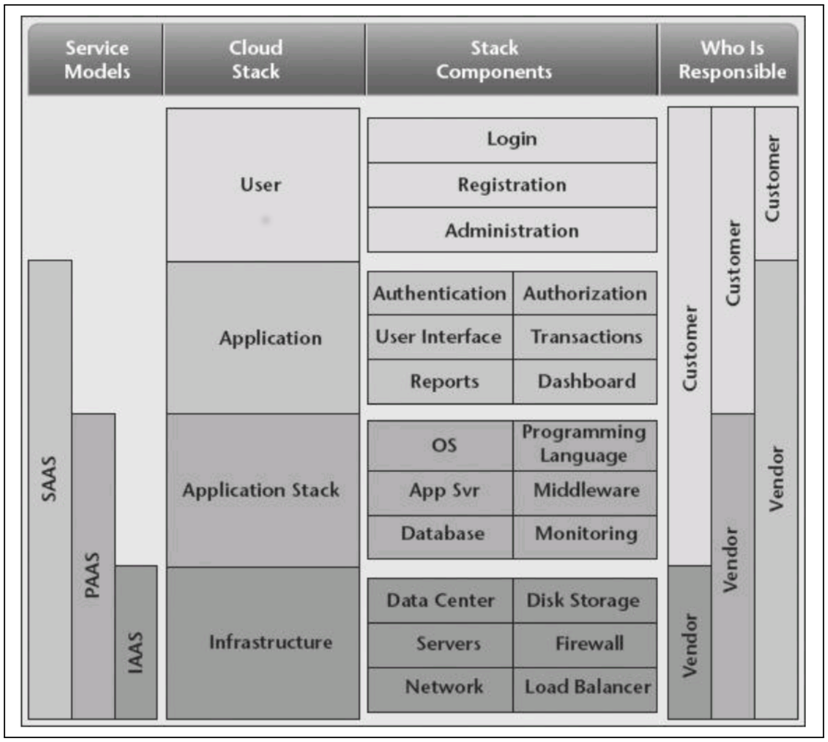
A definition given by other authors Buyya, Broberg and Goscinski (2010) emphasized cloud computing as a model to offer computing power via the network. In addition, the cloud uses powerful technologies, such virtualization that brings to companies an easier and fast service provision. Furthermore, as the computational resources (network, storage, user interface) are fully configurable and flexible.

Also, according to Sosinsky (2010), cloud computing presents two different classes of clouds: based on the deployment model and those based on the service model.

2.2.5.1 Service Model

A cloud can be presented as a stack, being designed as pyramid build by three layers (Iaas, PaaS and SaaS) recognized by the NIST as the cloud services. At the bottom it has the infrastructure which provides resources (network, storage, processor, memory) to the layers above. In the middle, it is present the platform layer which offers a customizable environment to development and testing tools. At the top of the stack are located the software layer, where the programs run and may host application to end users (Sabharwal (2013)).

The deployment defines the infrastructure organization. On the other hand, the list of services available by cloud vendors are the service model Sosinsky (2010). Such models bring a best understanding of the cloud system to users. Also, each company needs to analyze its demands at the time of the decision-making for a cloud. It happens mainly because the companies will depend on cloud computing power for their most important tasks.



Source: Kavis (2014, p. 46)

Figure 2.6: Types of Cloud Services.

- **Hardware as a Service (HaaS)** [1]

  HaaS is a different approach of cloud service. It is presented motivated by its growing and demand. A highly accepted cloud service model is HaaS because it can achieve high levels of cost-effectiveness. Therefore, in this model an IT provider offers hardware equipment (eg. Servers, printers, wireless routers), and is responsible for the maintenance and management. Buyya, Vecchiola and Selvi (2013)

  On the other hand, the corporations rent and pay fees to use the IT equipment, unless of spend a huge amount of money buying a high-cost hardware.

  An advantage for HaaS is the fact that some applications only work on this service model. Such applications that require direct access to hardware (eg CPUs cores , memory blocks) and does not work using others cloud service models (IaaS, SaaS, and PaaS) because they use additional abstraction layers (virtualization,

[1]Not recognized yet by the NIST as a service level. However is large used on the Market

management, APIs). Also, the hardware may be distributed over the world, on different geographical locations. Finally, it is possible to deploy a fast and reliable communication link between different data centers,where the transparency appears to the operational system as they were connected within a local network.

- **Infrastructure as a Service (IaaS)**

  According Sosinsky (2010) IaaS is the type of service model responsible to provide virtual machines and virtual infrastructure (computation, storage, and communication) related to hardware abstraction layer. Furthermore, Buyya, Broberg and Goscinski (2010) defined IaaS as the service that offers scalable and on-demand virtualized resources.

  Another complement is given by Williams (2010), which presented IaaS as an elastic capability which allows cloud providers to easily manage and offer computational resources. Also, it achieves a best hardware usage, where multiple virtual machines can run and share the computational resource(storage, CPUs, memory) even when are being executed on a single physical machine.

  IaaS Workload

  A workload often tests the overall capacity of a system to process a specific amount of tasks(SOSINSKY, 2010). It is a common way to measure and apply metrics of performance (Transactions Per Minute (TPM), load, throughput, latency, among others).

  Related to the cloud, the client request the resource to run the workload. On the other hand, the server reserves the amount requested following SLAs among the users. Also, IaaS layer, provides the demand of workload by the management of the virtualization and infrastructure.

  IaaS Aggregation

  The cloud workloads support as many users as possible. Often, several users share the same instances (using quotas) or a group of users. As a consequence, a group of users sharing the same virtual machine is named a pod. Several pods perform an availability zone (SOSINSKY, 2010).

Moreover, there is available another term related to cloud management, a silo can be defined as an isolated cloud. Such isolation often occurs using the network to segregate between different cloud characteristics (instances, users, location, among others)

- **Platform as a Service (PaaS)**

  As was presented by Buyya, Broberg and Goscinski (2010), PaaS offers an environment that allows developers to create and deploy applications. Furthermore, there are multiple programming models and specifics services that are available to build new applications(data access, authentication). One example of PaaS cited by the author is Google AppEngine that offers a scalable environment for developing and hosting Web applications.

  Additionally, the authors of Sosinsky (2010) emphasized PaaS as a flexible development environment for users to test, deploy and code the applications supporting several programming languages. The author also highlighted an important detail, the cloud provider extends and supports to users the infrastructure, operating system and third-party software. On the other hand, the client handles the application and control the development environment.

- **Software as a Service (SaaS)**

  According to Sabharwal (2013) Software as a Service aims to deliver desktop applications over the Internet. Thereby, the user does not need to install the application on his own desktop. SaaS makes possible to access remotely servers and run the programs.

  Also, Buyya, Vecchiola and Selvi (2013) presented Software as a Service as a model that brings to users access to software through a Web Browser. The user will just use his credential for accessing to software without the pay need for required licenses. Moreover, the applications can be customized to the client needs.

Another author, Sosinsky (2010) defines a SaaS as a complete environment that includes applications, management, and the user interface. Such environment is presented to the user through virtual machines and they are not required to care about other layers (Hardware, Virtualization, physical commodities). On the other hand, the user does not know where the application and services are hosted and running. In addition, Buyya, Broberg and Goscinski (2010) emphasizes another benefit of SaaS: this type of service model facilitates the development and testing of software by the organizations that provide them and decrease the demand for maintenance, and support.

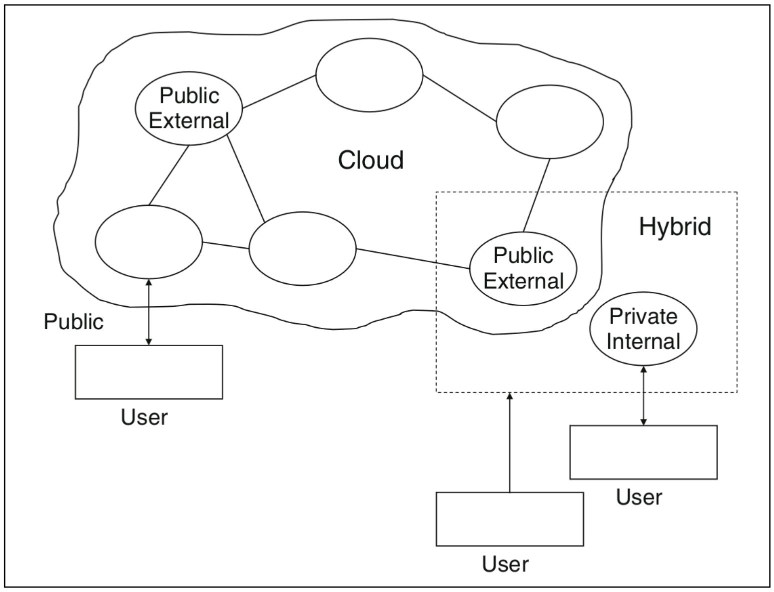
2.2.5.2 Deployment model

The deployment model is for indicating where the cloud is located and for what purpose. According with the deployment, a cloud can be classified as public, private, hybrid and community. The concepts of each one are presented in the following topics.

- **Public Cloud**

  A Public Cloud is defined by Buyya, Broberg and Goscinski (2010) as a cloud which is available to the general public that pay for the usage to a provider. The author uses the expression pay-as-you-go to describe the characteristic of this deployment model of cloud computing. Such expression means that cloud resources can be bought from others companies by the payment according how (the resources) has been used.

  Complemented by the author Williams (2010), has described a public cloud like a pool of computational resources and services brought to clients over the Internet. Such provision has been highly accepted by domestic and businesses users motivated by the practicality and cost-effectiveness.



Source: Furht and Escalante (2010, p. 7)

Figure 2.7: 3 Types of Cloud.

- **Private Cloud**

  As is presented by Sosinsky (2010) a private cloud is deployed and used by one organization. This cloud is hardly ever available to the general public. It is deployed in the internal data center using the company own hardware components.

  Williams (2010) defines private cloud as a cloud deployment within the company infrastructure. Such deployment requires a previously existing or acquisition of infrastructure because the cloud will run in the corporation hardware. Despite the infrastructure demand, a private cloud brings several advantages to companies (increase security and privacy levels, more flexibility, scalability, control, and customization)

- **Community Cloud**

  According Marinescu (2013), a community cloud is a type of cloud shared between organizations which have similar characteristics (eg., mission, security requirements, social interactions). The infrastructure is the key of the community cloud because the companies self-share their resources or split the payment of the investments with cloud vendors Williams (2010).

- **Hybrid Cloud**

  The hybrid cloud is composed by two or more clouds (private, public) that are bound together as a unit.

  The author Williams (2010) presented hybrid cloud as an interesting model because it combines the comfort for exploring the benefits of the public cloud to run applications and services. Also, it protects the vital data of the corporation inside its own data center (private cloud).

2.2.6 Service Levels for Cloud

The service levels are needed to emphasize what the cloud clients should expect to have available (eg., availability, reliability, and performance). In addition, according to Reese (2009), the providers in a traditional way offer to clients an SLA (service level agreement)

2.2.6.1 Availability

Availability is commonly related to how possible is to access resources in a time period. Currently, there is a trend for high availability of services. Applications should be available to end users 24 hours during the seven days of the week. However, it is a challenge to stay always online, to achieve high availability a system must have 99.99% of availability (REESE, 2009).

Moving this paradigm to a cloud, the virtual instances provided by public vendors often achieves 100% of availability. On the other hand, in the physical world if a server run into problems or worst than that, if a whole data center goes down it probably results in a moderated downtime period, even when applied redundancy techniques (RAID, backup, fault tolerance overall).

2.2.6.2 Reliability

The reliability is how much possible is to rely in your system integrity. Also, the availability impacts directly in the overall reliability due to the fact that an unexpected availability reduces the confidence of the system.

The author Reese (2009) emphasized also that the reliability is a consequence of how well the coding process has been done, and depends on the infrastructure equipment redundancy too (power, storage, trusted deploy).

2.2.6.3 Performance

The computing applications are still increasing the demand for performance. As a consequence, the computer have been grouped together to share resources (cluster computing, parallel processing, grid computing, distributed systems) and applications (clustering database, threading and process forking) to increase the performance (REESE, 2009).

A traditional deployment of applications for an acceptable performance is the usage of independent nodes (dedicated). It takes advantage of a load balancer that manages the requests for a undefined number of independent nodes.

On the other hand, the deployment can be performed through a load balancer within a clustered environment. In this model, the nodes are interconnected and shared between them the computational resources to answer the requests.

2.2.7 Cloud Architecture

An important aspect of every model or product is its architecture, which describes the features and shows the components involved. As expressed by Chandrasekaran (2014) the cloud technology follows this concept: "The cloud also has an architecture that describes its working mechanism. It includes the dependencies on which it works and the components that work over it". Finally, the architecture of a cloud system is shaped by four layers:

2.2.7.1 User/Client Layer

According to Chandrasekaran (2014), this layer is composed by the cloud clients and his/her devices. Moreover, often the users accessing the cloud remotely and the VM is running on an unknown location. Such connection depends on the network to have a successful response. Also, the user must have available at least one device (thin client, mobile device, computer) that runs a web browser to access the cloud.

2.2.7.2 Network Layer

The layer is responsible to grant the user access to the cloud. This layer is related to the external network used in public clouds, which may be accessed from every place in the world. However, it does not cover the network inside a cloud environment (CHANDRASEKARAN, 2014).

In addition, the entire cloud system is dependent of the network. It happens because the users have the flexibility of connection to receive on-demand resources over the internet. Also, the internet connection needs to supply a minimal bandwidth and reasonable latency in order to achieve the QoS.

2.2.7.3 Cloud Management Layer

This part of the cloud architecture is shaped by a software layer which controls the infrastructure and provides an efficient management to users. As defined by this softwares are used to Chandrasekaran (2014): "allow resource management (scheduling, provisioning, etc.), optimization (server consolidation, storage workload consolidation), and internal cloud governance."

The layer needs to grant and follow previously defined SLA rules. Such rules are created between clients and providers, either in public as a private cloud. The goals of these rules are for presenting which are the user's services available and ensure that it is offered properly by the cloud vendor.

2.2.7.4 Hardware Resource Layer

This layer is represented by the hardware and equipment. Often it is related to a data center and servers which process requests and provides on-demand computing power. There are a lot of complexities for managing several servers with powerful configurations. Despite the challenges, the hardware must be properly configured to offer resources following a QoS metric. It is important because this layer is under hard SLA rules (CHANDRASEKARAN, 2014).

Moreover, the cloud architecture layers above the hardware are highly dependent of this. As a consequence, a cloud provider has to attempt for high availability using redundancy and efficient algorithms.

2.2.8 Cloud Computing Related Technologies, Models and Paradigms

Cloud computing arose splitting several technologies (Virtualization, Grid Computing, utility computing, among others) and using its aspects to provide resources.

2.2.8.1 Web Services and Service Oriented Architecture (SOA)

Two important and base technologies of which enables cloud computing are Web Services and SOA. As presented by Furht and Escalante (2010), cloud services are built based on the Web services model for offering a wide variety of portfolio services.

Also, the Web services can bind with different applications that are being executed on different environments and exchanging information regarding applications between others. It is enabled by the a messaging interface where a classical usage example is in High availability systems (Buyya, Rajkumar and Broberg, James and Goscinski, Andrzej M., 2011).

On the other hand, the goal of SOA is to offer resources which meet the needs of the customers. The SOA standard includes the software provided as service. Such paradigm has been become largely used in cloud computing, offering solutions as services to ends users.

2.2.8.2 Virtualization

Virtualization is a technology which abstracts the hardware resources and makes easier to run applications. These tasks are performed by a part called Hypervisor, such as KVM and Xen. This technology is often used to aggregate several virtual machines within a single server which are managed and dependent of the hypervisor (ZHAO; SAKR; LIU; BOUGUETTAYA, 2014).

Moreover, the cloud computing tools takes advantage from the virtualization, using virtualization as a foundation. However, cloud computing goes far ahead from virtualization, and it is mainly related to additional services and smart resources management and allocation.

The author Marinescu (2013) presented the virtualization as the base of cloud computing. The virtualization allows users to share resources (memory, CPU cycles, I/O and communication bandwidth) among several virtual machines. The virtualization makes possible to isolate services on the same hardware and allows applications to migrate from one platform to another. There are several types of virtualization deployment, the main are described below.

- **Emulated**

  According to MATHEWS (2009), this platform is a set of hardware entirely simulated by the virtualization technology. The OS and application run without direct hardware interaction. This type is mainly used to test a software during its development.

- **Full Virtualization**

  As emphasized by the author MATHEWS (2009), this type is close to the emulated way. However, the main difference is the fact that the hypervisor allows the virtual machines to run instructions direct on the hardware.

- **Para-virtualization**

  In this type, the virtualized Operating System is modified in such a such way that the hypervisor exports a version of the physical hardware to the virtual system. The purpose of this is to easily accelerate the processing tasks (SOSINSKY, 2010).

- **OS-level virtualization**

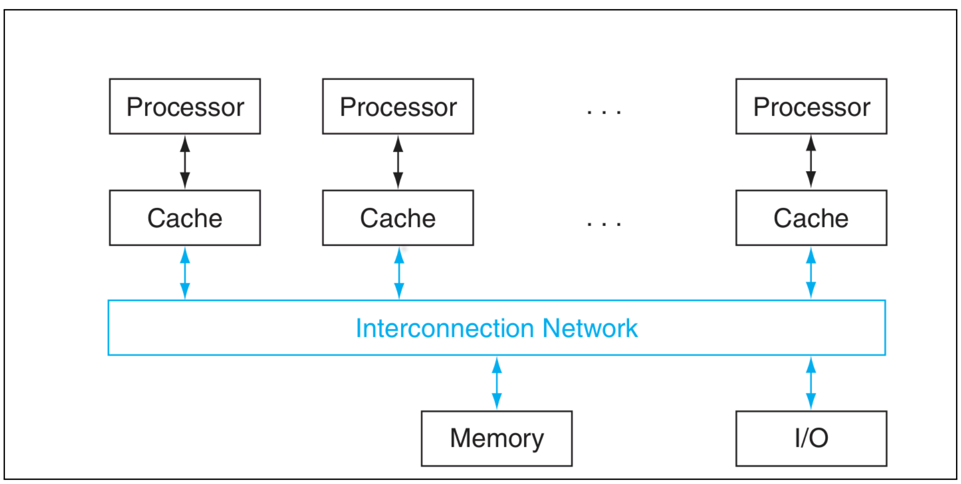
  In this model the virtualization is deployed direct in the host OS. Also, it introduces the concept of container environment which aims to provide programs and processes with an isolated environment to be run(MATHEWS, 2009).

2.2.8.3 Distributed Systems

After the improvement of the overall network performance, this model arose distributing the computational resources over the network. The concept of distributed systems is a set of autonomous computers connected in the same network. In this system, a distributed OS enables the resource sharing (hardware, software and data).

A quotation by the authors Vidyarthi et al. (2008) explains: "Distributed computing system falls in the category of distributed memory parallel architecture and is characterized by resource multiplicity and system transparency".

The distributed systems are only possible thanks to several other approaches (multi-task, multi-process, multi-core, shared memory, distributed file system) which are highly explored currently. The Figure 2.8 presents one example of shared memory on a multiprocessor system.

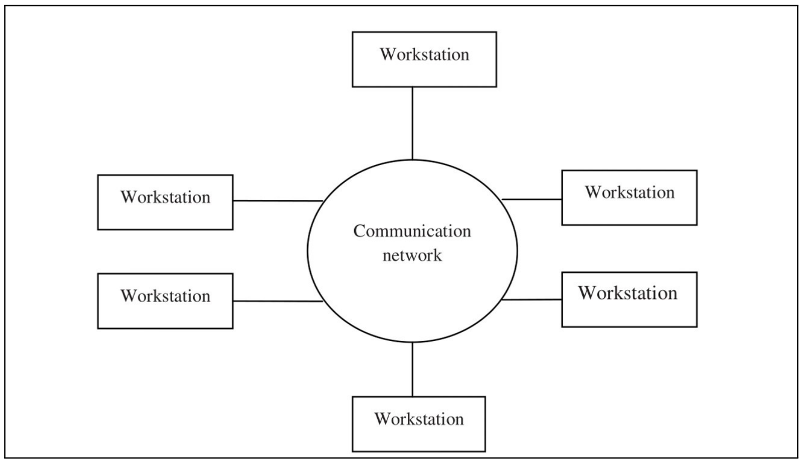


Source: Patterson and Hennessy (2013, p. 520)

Figure 2.8: Example of load distribution.

Since the distributed system arose several models have been emerged. The next Sections explain the majors.

- **Workstation model** The workstation model is connected in the same local network. Vidyarthi et al. (2008) defines this model as the simplest and most popular because the companies machines are not always in use. Then, the machines resources can be used running tasks and applications to other user or as a distributed collaborative servers (parallel execution).



Source: Vidyarthi et al. (2008, p. 19)

Figure 2.9: Workstation Distributed System model.

- **Cluster model**

  A cluster may be defined as a bunch of independent machines cooperating to solve computational processing task. Such computers are connected over a high-speed network Sterling (2002). When the cluster paradigm arose it brought massive improvements of the infrastructure organization and utilization. Actually, this model was one of the first efficient method introduced. Moreover, the cluster model is highly accepted and used in the era of cloud computing, either using to provide resources and offered as product by vendors (BUYYA; VECCHIOLA; SELVI, 2013).

  In addition, Vicat-Blanc et al. (2013) emphasizes servers for a cluster as a number of independent computers running in parallel aiming at overcome infrastructure limitations. This machines are organized inside racks and communicate through network protocols. It brings flexibility to add more servers according to the demand. As a result, cluster may surpass a supercomputer build with multiprocessors because a cluster is much more cheaper.

As the cluster became known, several types have been emerged (Commodity, Beowulf, constellations). This different types have their differences, often related to code openness, topology and hardware resources.

The commodity cluster is often provided by vendors that offer high availability and performance. The network should be dedicated to achieves high throughput rates. The bunch of computational processors and memory enables the cluster as a unity to handle a massive number of tasks and processes. Thus, the cluster provides a huge storage and the software used are hardly ever proprietary.

In contrast with the power of the commodity cluster, the Beowulf clusters are build of several interconnected PCs. It is an inexpensive and cost-effectiveness solution. Beowulf cluster also uses open source software operational system and middleware. A classical example is to build a cluster based on Linux OS. On the other hand, there are available vendors of Beowulf cluster, but they are often build on proprietary OS (Microsoft Windows).

Finally, the constellation cluster shows the paradigm of a huge number of processors (cores) on each node, and the number of processors is bigger than the number of the cluster nodes Sterling (2002). The tasks and programs processed on a cluster explore the system parallelism and shared memory to increase the overall system performance.

- **Client Server model**

  According to Tanenbaum (1995), in this model the computational resources and services are provided by several servers to users (clients) highly used in the field of distributed systems. The connection between the hosts are often done using the simple request/reply concept.

  The simplicity and efficiency of this model motivates the usage. If the system is properly configured when the client request a connection, it will always receive a response.

Moreover, the author Ghosh (2015) emphasized that sometimes a client is not reserved to only one server, such client may be using services or resources from a high number of servers.

- **Grid Computing**

  The Grid Computing model arose motivated by the demand for resources. Such resources were often used for processing a huge amount of data. In addition, this technology covers the distributed computing because of sharing resources over a network that results in the increase of the computational power(ZHAO; SAKR; LIU; BOUGUETTAYA, 2014).

  Despite the similarities between Cloud computing and Grid computing (distributed resources), the cloud goes ahead than Grid due to the usage of virtualization, which brings on-demand, flexible and scalable services.

2.2.8.4 High-Performance Computing

The High-Performance Computing (HPC) was introduced half century ago and had became a core technology under research and improve processes. The motivation for high-performance are the several scientific disciplines (eg., Chemistry, physics, meteorology, nuclear fusion), relying on computing technologies to meet their computational power need. Such demand, it is still pushing for increasing the computational power need and forcing the computing researchers to continue increasing the resources available. (FURHT; ESCALANTE, 2010).

As emphasized by Chandrasekaran (2014) in HPC based systems several processors (machines, cores or Hyper-thread) are grouped and they work together solving computational and scientific problems. Actually, the entirely infrastructure (memory, storage, I/O) is connected together using a software that makes possible the system operation (clusters). Also, the main paradigm of HPC are supercomputers (eg., multi-core and high clock processors, a huge amount of memory and high-frequency, high-speed network and disks).

The new generations of High-Performance computing system arose combining new technologies (virtualization and cloud). Such technologies offer an innovative way to provide and manage the resources and services. In this way, the resources management are in a dynamic and scalable way through the deployment of the virtualization. Also, the cloud paradigm controls the virtualization and brigs to end users this advantages (flexibility, scalability, on-demand services, among others).

In addition, as emphasized by Furht and Escalante (2010) the management via the cloud HPC tools goes forward than a traditional virtualization of a computer and it can be deployed in a cluster environment. In such deployment, the cloud tool handles the infrastructure management and works together with the software layers. Thus, using the cloud for HPC increases the efficiency, computational power and applications performance.

2.2.8.5 Virtual Appliances

A definition given by Buyya, Rajkumar and Broberg, James and Goscinski, Andrzej M. (2011), explains virtual appliance as a technology : "An application combined with the environment needed to run it". Currently, there is a trend to store virtual appliances in virtual disks. In such process, the challenges of the configurations and deployment are abstracted from users.

In addition, highly used operating systems and applications are offered by marketplaces as pre-build version. Such appliances offered are commercial or open-source supporting several flavors.

CHAPTER3: EXPERIMENTS AND RESULTS

This chapter presents the results and highlights the finds and conclusion of the research. The next Section presents related work, followed by the survey of IaaS cloud solutions. Also, the virtual pool performance is fully considered over a literature review. Finally, the conclusions, contributions and future works are presented.

3.1 LARCC

The LARCC [1] (Laboratory of Advanced Researches for Cloud Computing), as the description suggests, is a research lab related to cloud computing technologies. The core researches that have been done concern the performance of IaaS open source solution. So far, the most explored tools are OpenStack and OpenNebula, and they are related to the toolkits performance and features. However, there are available several others open source solution which is still under analyze by the research group. The research group is shaped by several universities (SETREM, PUCRS and Unipampa) which joined the project and provides human resources for the research. Also, there are several people involved and distributed among researchers, collaborators, and coordinators.

Additionally, the HiPerfCloud[2] project is being accomplished on the LARCC. This project is analyzing the cloud performance concerning a homogeneous compu-

[1] http://larcc.setrem.com.br
[2] http://hiperfcloud.setrem.com.br

tational environment. The HiPerfCloud project considers the aspects related to performance on cloud environments, over open source solutions. As the hardware provides computational power, and the virtualization abstracts the hardware and offer resources. Finally, the cloud frameworks manages the virtual infrastructure and provides resources as services to end users.

3.2 A SURVEY OF IAAS MANAGEMENT TOOLS

A survey has been performed motivated by the gap for precise information regarding the features (flexibility and resilience) of IaaS cloud tools. Contrasting with the comparison of tools done by Yadav (2013), in this paper we present the tools analysis and comparison, however, using a recommended survey methodology proposed by Dukaric and Juric (2013).

3.2.1 Survey Methodology

The methodology used was based on an unified taxonomy proposed by Dukaric and Juric (2013). Such methodology was applied in order to compare and classify the cloud components behavior. Also, several aspects and resources (hardware, network, virtualization) may impact on the performance results and QoS. As a consequence, well-done tools may not achieve the expected performance because of the impact of each environment characteristics or due to the fact of an internal inefficient performance management.

This survey compares IaaS open source cloud solutions over a taxonomy Dukaric and Juric (2013). Such number of tools have been increased lately. For this reason, the survey presents the solution available and related characteristics. Also, the solutions were ranked concerning each components and support for flexible deployments. The methodology is applied in order to highlight the similarities and the differences among the cloud platforms available to deploy a cloud virtual environment.

As presented and defined on Chapter 2, there are different cloud deployments models and types of services. In this paper, we focused the survey for IaaS open source solution. It enables the implementation of private, hybrid or public cloud, depending on the demand and on the tools flexibility. As defined by Serrano, Gallardo and Hernantes (2015), there are deep and significant distinction among private and public cloud providers. The next sections introduce this technologies.

The tools have been analyzed over a procedure that covers as most subjects related to robustness as possible. Over two independent comparison methods, one for flexibility and other for resilience, the tools were explored and checked if they covered or not each topic. Finally, the analysis and comparison have been done related to the scores of the tools support and the respective percentage of robustness (flexibility, resilience) achieved.

Additionally, in this survey only open source and free IaaS solution are explored and compared. It is an important contrast when comparing with the related works. However, is important to delineate a field and a criteria (methodology) in order to be able to go deep on the analysis.

3.2.2 Overview of IaaS Public Providers

When cloud computing emerged, companies started to offer resources over the Internet. Since 2006, when the first provider offered virtual machines the cloud is been growing. Currently, several vendors offer different services and resources to clients around the world. Consequently, IaaS became the most popular cloud product (as service). So, in this paper the major providers of IaaS public cloud have been selected and they are described below:

3.2.2.1 Amazon Web Services

Amazon web services (AWS) is a provider of cloud services (compute, storage,network) that offers to customers elastic and scalable infrastructure. The most common way to bill for services is via pay-as-you-go, the expense is related to the usage (BUYYA; VECCHIOLA; SELVI, 2013).

The amazon compute services deliver IaaS over the internet and its product is Amazon EC2. Amazon EC2 started as a service based on instances provision. In such instances, it allows the deployment of virtual machines and also servers.

Computer services offer templates (Amazon Machine Images) to install instances (EC2 instances). Such templates are based on a pre-selected amount of memory, number of CPUs, and storage capability. The instances may be accessed via command-line or using AWS graphical console. In addition, Amazon web services provide other services such as remote storage, virtual networks and messaging.

3.2.2.2 Google Compute Engine

The virtual machines provided by Google (2015) promise a high-performance, support to scalable environments, and achieve elevated security levels. The OS support offers flexibility to choose several distributions and availability to build large clusters on a high-speed network distributed around the world.

Also, Google engine enables load-balancing in such way that grant superior results related to performance and availability. In addition of easy deployment, Google offers support to different API and cloud managers. Finally, Google bills in a fairer way, based on minute level which may eventually avoid the payment of expendable computational resources.

3.2.2.3 Windows Azure

Microsoft Windows Azure is a cloud provider deployed on Microsoft hardware and offers services (compute, storage, networking) to customers around the world (BUYYA; VECCHIOLA; SELVI, 2013). Additionally, Azure supports IaaS and PaaS cloud models elevating the levels of flexibility and customization. Such flexibility allows users to deploy a cloud-based on any operational system and frameworks to all devices Azure (2015).

Also, Windows Azure promises advantages to end users, efficient and reliable storage system, high availability (redundant servers), complex networking (high bandwidth and throughput), and migration to other servers and applications. Additionally, it offers integration and easier management for windows administrators.

3.2.2.4 IBM SmartCloud Enterprise

IBM cloud IaaS IBM (2015) brings computational resources to deploy virtual or dedicated environments and execute production and applications tasks. IBM focus on control and flexibility to customers, combined with high availability. Finally, the IBM cloud focus on a balance between the enterprises priorities (management, software, security).

Moreover, IBM introduces IaaS instances as commodity services. Such concept, brings the idea of cloud as customizable and flexible provider of computational power to users.

3.2.2.5 HP Enterprise Converged Infrastructure

HP Helion cloud is a solutions based on code openness. This advantage is possible by joining and being an important contributor of the open source community projects (OpenStack, Cloud Foundry). For this reason, the HP cloud delivers products and services with exclusive features to its customers HP (2015).

In addition, HP emphasizes open source and freedom is the future path of the software. The cloud supports several deployment models (public, hybrid, private) by the integration among cloud tools. Moreover, HP supports Windows and Linux via command line or dashboard interface.

3.2.2.6 Locaweb

The Locaweb (2015) is a Brazilian cloud provider. Among several solutions presented on Locaweb portfolio, we are considering only solution related to IaaS. Locaweb offers the Cloud Server Pro, which is a product/service of computational resources to end users over the internet. Additionally, it offers several server flavors, segregated on memory amount, number of vCPUs and disk space.

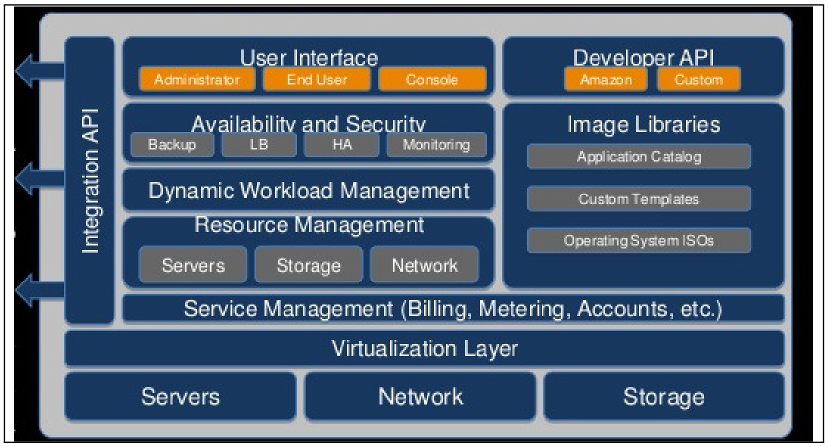
3.2.3 Open Source IaaS Tools

The infrastructure management tools are developed aiming at a best hardware usage and management.

3.2.3.1 CloudStack

CloudStack was developed by the start-up VMOps and changed its name to Cloud.com in 2008. A part of source of the CloudStack was released in 2010 and, in 2011 the rest of the source was released by the company Citrix, which bought Cloud.com. After that, CloudStack was submitted to the Apache Incubator. Apache CloudStack was graduated by the incubator in 2013. Cloudstack (2015a) is a project that develops open source software, which allows the deployment of public and private IaaS clouds. Apache CloudStack is manager by Project Management Committee

(PMC).



Source: Apache (2015)

Figure 3.1: Cloudstack architecture.

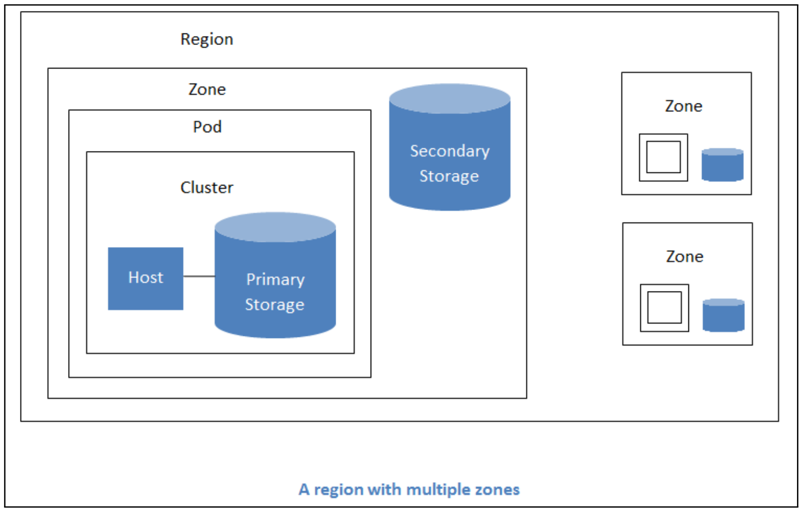
The Apache CloudStack is based on Java and presents the following features:

- Supports several Hypervisors as guest OS.
- Supports integration to Amazon S3/EC2 API.
- Dynamic storage management.
- Full and complex network compatibility and configuration.
- Users management and accounting for services and quotas (network, storage) via a UI.

According to Sabharwal (2013), the CloudStack management model separates the parts necessaries to build a cloud environment.

A zone is the high level class isolated from other zones (other owner). A classical zone is configured for each data center, or it separates a server into zones to isolate users and resources.

Inside a zone, there are at least one pod (the second-level hierarchical which has the cluster management charge). The cluster is the next class, shaped by virtualized server working as slaves.



Source: Cloudstack (2015a)

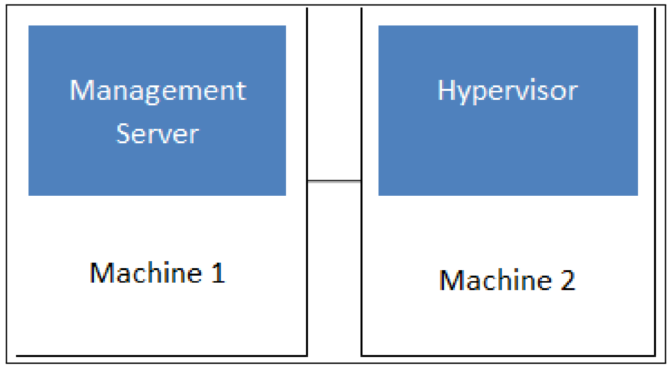
Figure 3.2: Cloudstack infrastructure overview.

The storage on CloudStack must have primary server that hosts the VMs and is mandatory at least one for each cluster. Such server has available a huge disk space to supply the users demand. Also, it supports local and distributed storage architecture. Finally, the file systems supported in CloudStack is a wide variety (SAN, iSCSI, NFS, EXT among others).

On the other hand, the secondary storage holds the ISO images, templates and snapshots. The secondary storage supports different file systems such as NFS and swift (OpenStack).

The management server is the main part of the cloud environment. It has the most important functionalities such: web interface, APIs management, Network controller, storage management, among others. On the other hand, in a classical approach, is created a slave host which provides its computational resources to the management server control and deploy instances. Such host has installed a hypervisor and a cloud-agent. Also it accesses computational power over the network.

Finally, CloudStack is based on java and promises several advantages inside its features. It is compatible with the most important current Hypervisor, offers to user and admins management and UI. Also, robustness for the network and stor-



Source: Cloudstack (2015a)

Figure 3.3: Cloudstack deployment architecture overview.

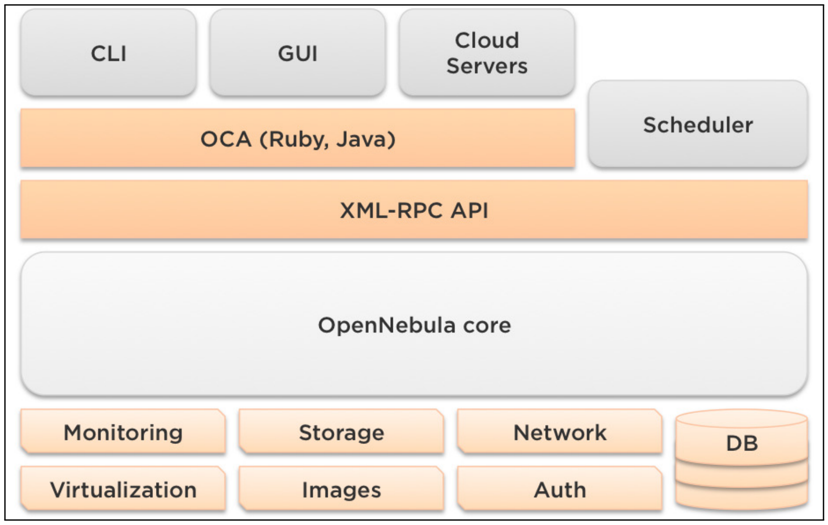
age (primary, secondary, and templates ISOs and snapshots). Further, CloudStack can easily allow the deployment of a public or a reliable private cloud over the multi-tenancy and native API.

3.2.3.2 OpenNebula

OpenNebula started as a research project back in 2005 by Ignacio M. Llorente and Rubén S. Montero. The software had the first public release in 2008. The evolved was made by open-source releases and currently is an open source project.

According the OpenNebula (2015a), the OpenNebula project pursues the following objectives:

- Deploy in a easier way a powerful and customizable virtualized cloud.
- Support to users and developers with options of cloud and users interfaces, which may create an elevated level of customization and components.
- Grant a software stable and reliable.
- Improve the data center management tools and the quality of services.
- Avoid errors in the project(poor code).



Source: OpenNebula (2015a)

Figure 3.4: OpenNebula infrastructure overview.

- Also, OpenNebula project aims for support and collaborate with the parts involved in the project deployment(developers), test (users) and other Open Source technologies

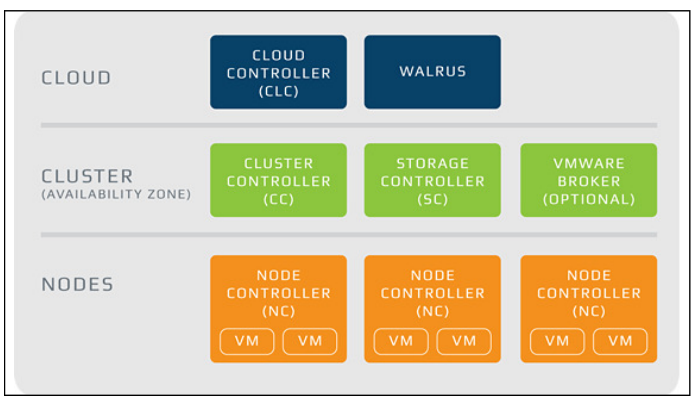
In addition, the lowest OpenNebula level (below OpenNebula's core) has several drivers. The transfer drivers are needed in order to control the storage system (NFS, iSCSI). Other drivers are the virtual machine managers (eg., libvirt), which are used to perform the Hypervisor jobs and control de virtual instances. The last drivers are the information drivers for getting data of the instances. Such files are transfered between the cloud Frontends and hosts over SSH.

Another concern of the OpenNebula solution is performance, the code is written in an optimized C++ in order to improve the performance and offer additional flexibility. Also, OpenNebula supports a hook, which allows the cloud administrators to automate custom scripts and tasks. The author of Toraldo (2012) highlights the opennebula security concerns, the host only communicates over encrypted connections using Secure Socket Layer (SSL) and the networks are segregated through the firewall ebtables.

The core of OpenNebula enables the easy and automated private cloud deployment (TORALDO, 2012). Inside such layer, several components and services run. The user manager controls everything about the tenancy (users, VMs, quotas, volumes, networks, among others). Further, the image management takes control of the VM disks.

The network management is used to look for the virtual network usage and control, as virtual machine and service management. Moreover, the OpenNebula core has also the infrastructure, storage and information management in order to improve the resources usage and availability. Another advanced approach of OpenNebula is the scheduler, which allocates VMs on nodes following a resource-aware policy (load distribution and balance). Finally, the User interface (UI) offers several command-line tools and graphical user interface (Sunstone) to enable the easy management.

3.2.3.3 HP Helion Eucalyptus

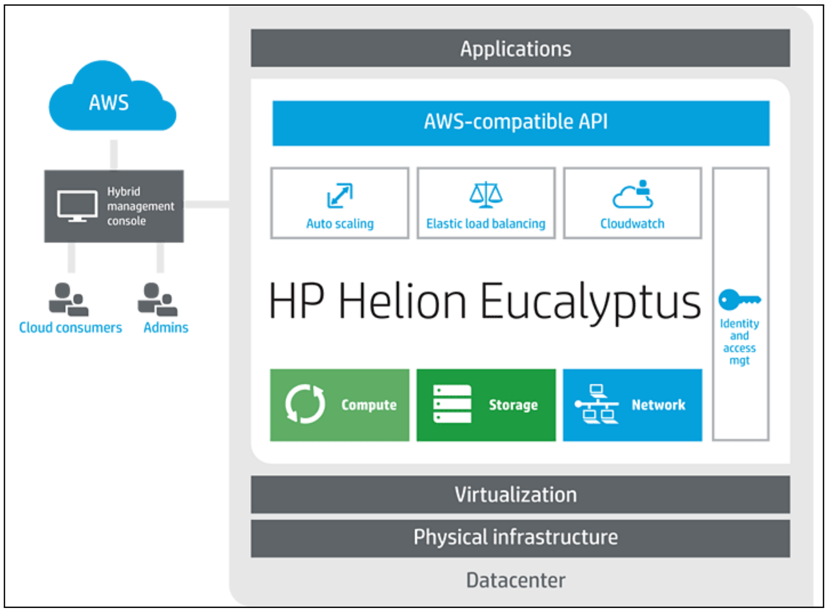


Source: Eucalyptus (2015a)

Figure 3.5: Eucalyptus components.

Eucalyptus (Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems) to its official website Eucalyptus (2015a) may be defined as an open source package to deploy Private and Hybrid Cloud. Looking for details, it focus on Infrastructure as a Service (IaaS) aiming to improve the management and services provision.

Such cloud technology allows to build a private cloud by using the heterogeneous hardware and to be an Eucalyptus (2015a) "*elastic resource pool that can dynamically scale up or down depending on application workload demands*" . Furthermore, Eucalyptus emphasises the benefits of the cost saving and more control of the corporation's data.



Source: Eucalyptus (2015a)

Figure 3.6: Eucalyptus architecture overview.

Also, it provides APIs compatibility with Amazon cloud (AWS). Eucalyptus is build aiming for scalability of its services. The storage brings flexibility to increase the number of Hard drives and storage server dynamically. Additionally, the network offers a high customization level, supporting several virtual topologies, protocols, routers and dynamic growth. According Marinescu (2013) the Eucalyptus main architecture components are described in the next paragraphs:

The node controller is done be by services that run on each server or node dedicated to host virtual instances in such way that provides the resources and manages the node (BAUN; KUNZE; NIMIS; TAI, 2011).

Also, the cluster controller runs on the server side. Such server controls and balance between requests and the node controllers (slaves).

The hosts responsible for data storage are the storage controller and storage services. The storage controller offers virtualized hard drives to the virtual machines and users. On the other hand, the storage services enables the storage objects.

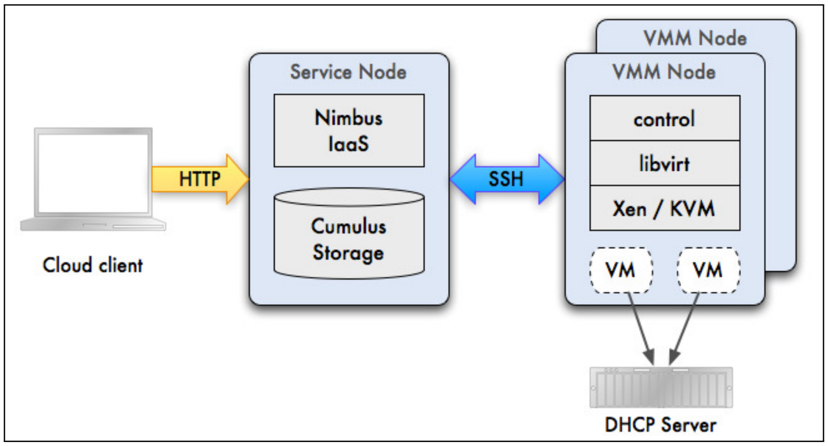
3.2.3.4 Nimbus

According to Nimbus (2015) this tool can be described as a solution for provide Infrastructure-as-a-Service cloud over clusters and is focused on deployment for the scientific world.

One of the main goals is to allow cloud providers to deploy and offer private and public clouds. In such deployment, the users are able to instantiate virtual machines powered by the cloud servers. The Nimbus storage and images repository is managed by Cumulus, which enables the storage provision based on quotas. It aims for scalability and easy configuration of multiple storage deployments.

In addition, Nimbus allows cloud developers to customize their cloud deployment. Such jobs can be expressed by the variety technologies (Hypervisors, user interfaces, schedulers, among others) supported for deploy a cloud environment. Also, the flexibility covers the possibility of extend a previously existent cloud.

As presented by Mahmood (2014), the Nimbus platform architecture has several components. The cloud client is who uses the virtual resources over a command line interface. Nimbus interfaces are frameworks for integration with the amazon EC2 cloud, while the Nimbus service is interaction for control the tasks. Also, the Nimbus has the workspace resource manager for control over SSH the VMMs which interacts and manage between the user and instances requests. Finally, Nimbus uses a context broker in order to offer for end users advanced and automated resources, such



Source: Nimbus (2015)

Figure 3.7: Nimbus deployment.

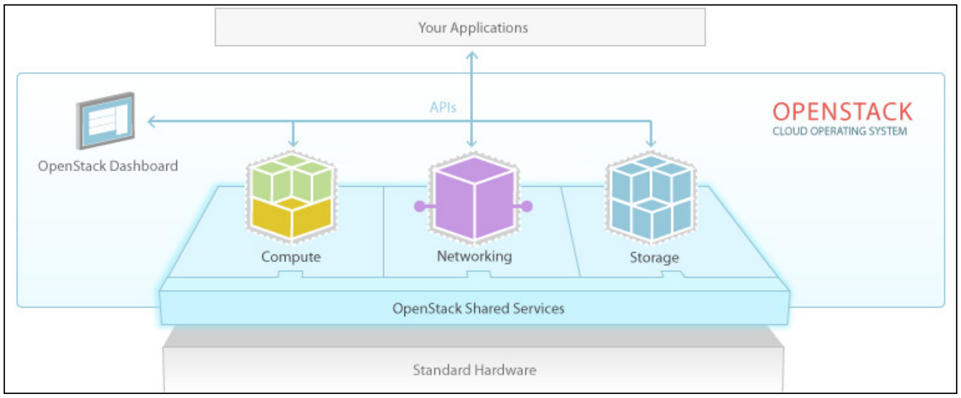
as launch, control, and edit virtual instances and applications (CAMPESATO; NILSON, 2010).

3.2.3.5 OpenStack

OpenStack is an Open Source aggregated software solutions, for cloud operating system, which supports private and public clouds. The project began in 2010 launched by Rackspace and NASA after others big companies joined the project. Firstly, OpenStack aimed at support the interoperability of cloud applications and makes possible to corporation build their own cloud using its infrastructure. Also, it promises a best usage of the hardware, and the support for the customers build the cloud OpenStack (2015a).

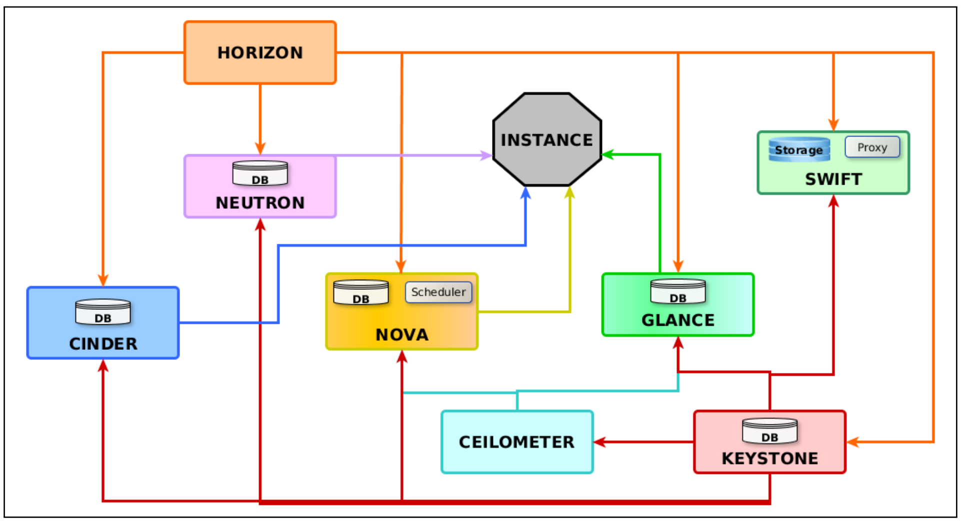
Furthermore, OpenStack aims to be a strong choice, because its software supports cloud providers (offer services over the Internet) and solution for companies to build a private cloud. Thus, OpenStack emphasizes its flexibility to all possible scenarios and demands Kumar et al. (2014).

The OpenStack compute is the main component of an IaaS cloud. Moreover, the compute interacts with other components in order to authenticate, OS images and



Source: OpenStack (2015a)

Figure 3.8: OpenStack overview.



Source: OpenStack (2015b)

Figure 3.9: OpenStack conceptual architecture.

usage management. The OpenStack compute management is performed by the Nova component OpenStack (2015b) often it is installed on the OpenStack management server. Also, in the cloud controller and the keystone are aggregated too (FIFIELD; FLEMING; GENTLE; HOCHSTEIN; PROULX; TOEWS; TOPJIAN, 2014).

The storage is an important part of the OpenStack system. It might be deployed using two different types, LVM block storage using the Cinder component and object storage through the swift component. Furthermore, the volumes may be local or distributed over the network.

Finally, the OpenStack networking uses the neutron API (DENTON, 2014) and supports several plugins (eg., OpenVSwitch), drivers and services in order to perform the connection and process the data transfers (attach interfaces, establish connections, VPNs, IP releases).

3.2.3.6 OpenQRM

The OpenQRM Community Edition OpenQRM (2015a) is an open source cloud tool administrated under the GPLv2 license. OpenQRM has and enterprise edition which is focused in corporations and combines the support and wonder about the releases of the open source edition.

OpenQRM is defined as an open source data center management and cloud computing. It main advantages are the flexibility and support to the most important virtualization technologies. Also, OpenQRM aims for high availability by the virtualization and redundancy of the services (Storage, virtualization, network).

As defined by Vaquero (2012), OpenQRM has been built based on a modular architectures. The system supports the deployment of virtual machines and advanced management of the computational resources (eg.,monitoring, performance, scalability).

Additionally, OpenQRM cloud Enterprise edition can take advantage of several hypervisor (KVM, VMware, VServer and XEN). While the community edition supports only KVM. On the other hand, the storage supported is more flexible on both editions, it supports a complete bunch of storage file systems such as NFS, iSCSI, NetApp and LVM.

Finally, OpenQRM presents one of the easiest cloud deployment due to the plugin API based installation. In other words, after a basic installation and dashboard access, the cloud administrator can easy customize the environment by one click plugins enable. Also, another interesting approach powered by OpenQRM is the multi-language web interface and cloud portal.

3.2.4 Related Surveys

A paper authored by Prodan and Ostermann (2009) is focused in a terminology to classified the cloud providers and its instances. Conducted a survey of the providers analyzing eight important elements of a taxonomy: service type, resource deployment, hardware, run-time tuning, business model, middleware, and performance.

Their survey is based on a restricted list of cloud providers because of the paper space limitations. The providers were selected based on the information and details that were available in their websites. The paper compared the instances types available and the cost associated with performance, based on Linpack benchmark.

Their conclusion introduces the three most important elements for cloud research: provisioning of Service Level Agreements as utilities, open and interoperable middleware solutions, and performance metrics for high-performance computing applications.

Also, another survey, by Endo et al. (2010) presents and discuss open source solution for cloud. Motivated by its advantages, cloud is the next generations in terms of computational resources and services to corporations.

The solutions studied were Xen Cloud Platform, Nimbus, OpenNebula, eucalyptus, TPlatform, Apache Virtual Computing Lab (VCL), Enomaly Elastic Computing Platform. The study points out the need for the standardization of the cloud services and tools. Also it highlighted OpenNebula by being the first to deploy policies for resource management because it is a trend.

Additionally, a paper of Wu, Baasha and Karwa (2010) surveyed and gave a taxonomy between three open source source cloud IaaS (Nimbus, Eucalyptus, Open-Nebula). It compared the work flows and behaviors of the tools running map reduce.

The authors concluded that the architecture, system setting, and functionalities between the IaaS tools are close. The study realized how important the differences between the different ways for user interaction (web portal, command-line) and the importance of monitoring the hardware resources (memory, CPUS).

In addition to several cloud surveys, the study Dukarić and Jurič (2013) analyzed only IaaS solution, either public providers as open source source solutions. This study presents a conceptual taxonomy for cloud tools. Such taxonomy was built considering what the cloud frameworks supposed to support. The results show how different the solutions are and the concerns for a careful consideration when chosen the most suitable solution.

The paper Thome, Hentges and Griebler (2013) shows a comparison among the open source IaaS tools: Eucalyptus, OpenNebula,OpenQRM, OpenStack, Cloud-Stack Ubuntu Enterprise Cloud, Abiquo, Convirt, Apache Virtual Lab e Nimbus. Also, considering the functionality and deployment model of the tools.

The study was motivated by the difficulty to choose the most appropriate tool for the companies. The authors concluded that OpenQRM was the most complete tool, supporting a large variety of tools and environment for deployment. In addition, Eucalyptus, OpenNebula and OpenStack have had good review results because the authors emphasized that are complete solutions too.

The survey taxonomy and methodology proposed by Dukaric and Juric (2013) is modular and consider a fair approach of the tools. For this reason, this methodology is used in order to analyze the cloud solutions. In other words, further than explore only the tools flexibility, we are also looking for the resilience of the frameworks and its underlying technologies.

Also, another important aspect of this survey contrasting with the studies already done is the fact that we only compare in this paper open source and free IaaS solution. It is distinct of the papers which compared public cloud providers (Dukarić and Jurič (2013), Prodan and Ostermann (2009) and Endo et al. (2010)). Also, the main difference with the paper that compared open source solutions for cloud (Thome, Hentges and Griebler (2013) and Wu, Baasha and Karwa (2010)) is the fact that this paper aims to update the knowledge of the solutions available (tools) and use methods for analyze the support for robustness on each cloud IaaS tool. The table 3.1 summarizes the related surveys.

Survey	Tools/Providers	Objective
Prodan and Oster mann (2009)	Public cloud providers	Classify the cloud providers
Endo et al. (2010)	Xen Cloud Platform, Nimbus, OpenNebula, eucalyptus, TPlatform, Apache Virtual Computing Lab (VCL), Enomaly Elastic Computing	Present and discuss open source solution for cloud
Wu, Baasha and Karwa (2010)	Nimbus, Eucalyptus, OpenNebula	Taxonomy and comparison
Dukaric and Juric (2013)	OpenNebula, Eucalyptus, OpenStack, Nimbus, AWS, Microsoft cloud, Vmware vCloud	Present a conceptual taxonomy for cloud and compare solutions.
Thome, Hentges and Griebler (2013)	Eucalyptus, OpenNebula,OpenQRM, OpenStack, CloudStack Ubuntu Enterprise Cloud, Abiquo, Convirt, Apache Virtual Lab e Nimbus	Tools comparison concerning the functionalities deployment model

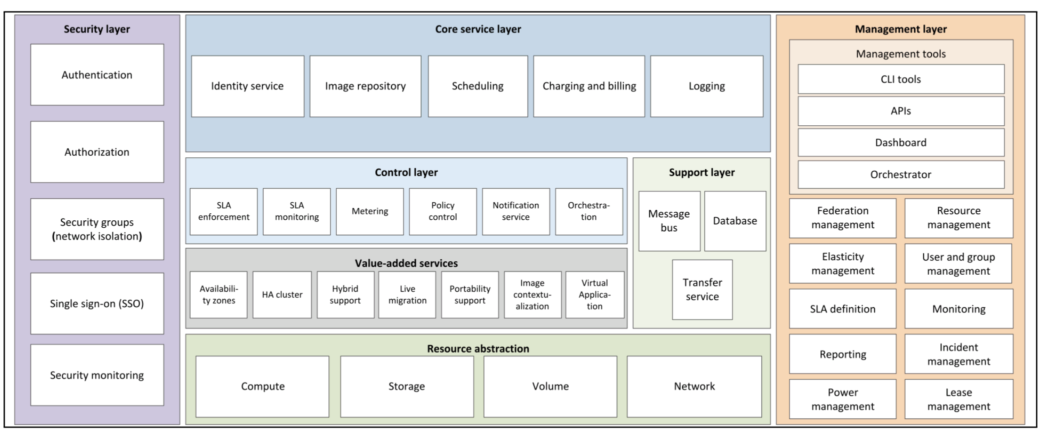
Table 3.1: Related Surveys.

3.2.5 IaaS Tools Taxonomy

Taxonomy is the science of categorization, or classification, of things based on a predefined system and contains a controlled vocabulary with a hierarchical tree-like structure Liu et al. (2011).

A paper authored by Dukaric and Juric (2013) emphasized an unified taxonomy for cloud frameworks. Following this concept, it is possible to find out important points about cloud taxonomy and comparison between solutions. Using such methods, it may help to classify the tools studied according to its characteristics and components. For finally, provide information about the tools and the most indicated environment to each deployment. Furthermore, are considered only some specific layers related to IaaS: virtualization abstraction layer (compute, storage, network), management layer (monitoring, power management, UI). Also, some specifics topics are considered for comparison (OS and Hypervisor compatibility and support, among others) which are added because the research has special goals (robustness) that need to be taken into account.

Also, the unified taxonomy proposed in the paper of Dukaric and Juric (2013) is used in this paper as a methodology to analyze and compare the cloud IaaS solutions. Such taxonomy is divided on seven layers (core service layer, support layer, value-added services, control layer, management layer, security layer and resource abstraction) as a conceptual and fundamental architecture for IaaS frameworks to fill. The structure organization based on layers presented by the author of the unified taxonomy (DUKARIC; JURIC, 2013) are described and its parts detailed below:



Source: Dukaric and Juric (2013)

Figure 3.10: IaaS proposed taxonomy.

3.2.5.1 Resource Abstraction Layer

The components (compute, storage, volume, and network) within the resource abstraction layer are the lower in the architecture of a cloud solution. Often, such resources are provided using virtualization that interacts directly with the hardware in order to provide computational power.

The compute service is an abstraction layer only possible by using virtualization technologies in such way that this component offers resources (CPU, memory) and manages (deploy, delete, start, stop, shutdown, etc.) to the layer above (virtual machines). These tasks might be performed using different technologies and ways, always depending on the Hypervisor support. The most common compute approach is using daemons or specifics API.

The storage service idealized by cloud solutions offers to users a reliable and scalable option to store huge amounts of data. Looking to open source, each cloud tool storage model is unique.

The storage performance is vital for the overall system because it impacts directly in the capability and time response for process real-time requests. In addition, the demand and operations disks (read, write, re-read, re-write, random read, random, write) may vary according to each application characteristics. Consequently, the tools supposed to support as much file systems (eg., ext3, ext4, NTFS, FAT), disk formats (QCOW, LVM, raw file) and distribution models (GFS, NFS, LVM, etc) as possible. On the other hand, it may become complex to handle and manage such storage robustness. Either way, the proper act of store data is performed in volumes which are controlled by a storage system.

As defined by Dukaric and Juric (2013) the network may be: "the most complex module of any Cloud Computing framework". Such factor may be justified due to the complexity of properly configure and achieve a regular network performance results. The cloud topologies are often distributed over a high-performance network in order to increase the overall computation power. As a consequence, there is a trend for improve the network throughput and latency.

The cloud tools adopted several variations of the network architecture, which goes from a simple client-server model to a complex high-performance powerful distributed cluster topology. The cloud drivers need to handle and abstract the complexities related to interfaces, IP, MAC addresses, bridges, routing, and VLANs. Also, the cloud platform sometimes uses customized internal network controllers or may use open network plug-ins, such as OpenVSwitch and Neutron.

3.2.5.2 Core Service Layer

A phrase written by the author Dukaric and Juric (2013) defines the core layer as: "The main part of the proposed architecture" and is shaped by five parts (identity service, scheduling, image repository, charging and billing). As the name suggests, the identity service interacts with other cloud components, offering an additional security layer authenticating and controlling users. In contrast, the scheduler deals with the resources and hosts allocation to run instances. Also, the scheduler is an important key enabling on the cloud operations jobs such as VM creation, load balancing and consolidation components.

Furthermore, the image repository offers to cloud user a pre-built images catalog. Such service often works in such a way that when a virtual machine is deployed the image is copied into volumes to run the OS. Finally, the charging and billing components, handles the logs, and users access to billing in a customized way.

3.2.5.3 Support Layer

This layer works as middleware that controls and communicate with other layer and services. The conceptual components of the support layer are message bus, database and transfer services. The message bus act as "central station" controlling and sinking the interactions between services and APIs. In contrast, the database deals with the storage of settings and real-time operations (instances, networks, images). On the other hand, the transfer service handles the exchanges during the VMs deployments. Sometimes transferring in the same server or the most complex act when is required the transfer between different hosts or clusters.

3.2.5.4 Security Layer

As emphasized by Chandrasekaran (2014), security is a big challenge in cloud system and involves several aspects, such as data, data center, virtualization, network, among others. As a consequence, the unified taxonomy proposed by Dukaric and Juric (2013) gave a special attention to security. The layer is divided into five indispensable aspects: authentication, authorization, security groups component, single sign-on and security monitoring. The authentication is necessary to check if the access is allowed or not. Such process often requires a user and password to grant the access to a cloud pool. Also, authorization is performed after authentication, where the user follows rules previously defined based on what he/she can execute or not within the cloud environment.

Besides, a security group can be defined as several users who share resources (instances, volumes, networks, compute) and a programmable amount of computational power (quotas). Also, the single sign-on happen when the credentials are shared among several interoperable systems in such way that the user only authenticates in one and is possible to access another independent system without authentication. Finally, the secure monitoring covers techniques aiming at protect the cloud environment. Such monitoring watches the cloud resources (power, users, access, among others) using the artificial intelligence concept.

3.2.5.5 Management Layer

The proposed management layer inside open source IaaS frameworks is another really important part. The taxonomy split the entire layer into eleven different components: Management tools, federation management, elasticity management, resource management, user and group management, SLA definition, reporting, monitoring, incident management, power management and lease management.

Management tools are related to interactions with users. Several cloud solutions support admin and end user graphical user interface (GUI), which is also known in the cloud field as a dashboard. Such interface is commonly accessed over a web browser and provides control of the cloud environment. Another more powerful interface is the CLI (command line interface), which is often used by cloud architects to run tasks in a more flexible and faster way. Also, the APIs supported by cloud tools may offer an interface to control the infrastructure too.

Two components that are related to multi-cloud are the federation and elasticity management. The elasticity refers to the capacity increases or decreases the number of servers, VMs, and users without compromise the availability and QoS. Also, the federation management is the act of manage multiple clouds using a unified method. Such techniques facilitate the users and groups management because guarantee that the resources meet the users demands.

Moreover, the reporting component is responsible for report errors and prevent unavailability of the overall system. A common example is when the monitoring system realizes about a situation and reports the issue. The incident management can be defined as the fact of prevent such issues and explains what should be done in such cases. Finally, the lease management is the ability of control the tenancy for cloud clients (FIORE; ALOISIO, 2011), (ROVEDA; VOGEL; GRIEBLER, 2015).

On the other hand, the SLA definition covers the contract of services and the needed QoS, which may be contracted and the act of SLAs grant such quality. In contrast, the power management is related to the controls and monitoring act of the energy consumption and the plan of reduce such impact. Thus, the resource management covers the best usage of the computational power provided by hosts of the cloud and also goes through the control of virtual machines usage.

3.2.5.6 Control Layer

This layer, as its name suggests, is related to the control of the characteristics of an IaaS framework. The components inside this layer are SLA enforcement, SLA monitoring, metering, policy control, notification service and orchestration component. The issues related to SLA are applied only for public cloud, for example, if the goal is to build a private cloud inside an enterprise domain using open source tools, the SLA is not a big deal. In contrast, when we need to contract computational resources from a cloud provider is highly recommended to have a SLA agreement in order to guarantee the QoS. For such process, there is a need for monitoring system, which analysis the availability and supports complete reports.

In contrast, resource metering is an efficient way to follow the cloud usage and bill according to the usage. The policy control are rules applied to the usage of a cloud environment. Lastly, the notification service is enabled aiming for improve the control of actions and over unexpected situations. Finally, the orchestration is a platform that facilitates the management of a cloud environment. The most common approaches are visual interface improvements, integration, and easy instances deployment.

3.2.5.7 Value Services

Following the proposed taxonomy by Chandrasekaran (2014), this is shaped by the additional and optional components of a cloud architecture, such as availability zones(redundancy across different locations), high availability(reliable cluster), hybrid cloud support (connect to another type or cloud service), live migration (transfer instances across hosts), portability (flexibility over several disk formats, hypervisor, APIs), image contextualization (images standard for deployment) and virtual application support (deployment of suitable applications enabled).

3.2.6 IaaS Tools Comparison

Since the rise of cloud computing, it has been a hot research topic. The majority of cloud frameworks are robust and complex technologies. While the cloud applications and usability is still being explored by the scientific field, as presented by Peng et al. (2009). However, the IaaS cloud tools architecture and flexibility for a customizable deployment are unclear or unknown. For this reason, in this thesis we present a taxonomy and comparison among the open source solutions available to deploy an IaaS cloud.

Another important aspect of the suitability of cloud tools is the robustness. The study of Srivastava, Ciorba and Banicescu (2011) fragmented robustness into two approaches, flexibility and resilience.The flexibility is how customizable and wide the solutions are, while resilience refers to the levels of elasticity that may be achieved.

3.2.6.1 Flexibility

As presented on Table 3.2, the IaaS solutions have different features and goals. When the character "/" appears it means that in this layer or topic the solution do not support or is unknown such support. Also, the word "internal" means that the tool support this, but is unknown which component performs this job or is inside the solution core. Additionally, "external" has been presented when the solution can easy achieve this demand just by adding an external and a fully compatible component or framework.

The Table 3.3 shows the compatibility of the IaaS open source solutions for flexibility. Regarding the resource abstraction layer of the tools, which is the closer abstraction of the virtualization layer. The OpenNebula uses the Oned for provide computer resources (CPU, memory). Looking on eucalyptus compute, there is a unclear

Support	OpenNebula	Eucalyptus	OpenStack	Nimbus	CloudStack	OpenQRM
Resource abstraction layer						
Compute	x	x	x	x	x	x
Storage	x	x	x	x	x	x
Volume	x	x	x	x	x	x
Network	x	x	x	x	x	x
Core service layer						
Identity service	x	x	x	/	x	/
Scheduling	x	x	x	x	x	x
Image repository	x	x	x	x	x	x
Charging and billing	/	/	x	/	x	/
Logging	x	x	x	x	x	x
Support layer						
Message bus	x	/	x	/	x	/
Database	x	x	x	x	x	x
Transfer service	x	x	x	x	x	x
Management layer						
Resource management	x	x	x	x	x	x
Federation management	/	/	/	/	/	/
Elasticity management	x	x	x	/	x	/
User/group management	x	x	x	x	x	x
SLA definition	/	/	/	/	/	/
Monitoring	x	x	x	x	x	x
Reporting	x	/	/	/	/	/
Incident management	/	/	x	/	x	x
Power management	x	x	x	x	x	x
Lease management	x	x	x	x	x	x
Management tools						
CLI tools	x	x	x	x	x	x
APIs	x	x	x	x	x	x
Dashboard	x	x	x	/	x	x
Orchestrator	x	/	x	/	x	/
Security layer						
Authentication	x	x	x	x	x	x
Authorization	x	x	x	x	x	x
Security groups	x	x	x	x	x	/
Single sign-on	/	/	/	/	x	/
Security monitoring	x	x	x	x	x	x
Control layer						
SLA enforcement	/	/	/	/	/	/
SLA monitoring	/	/	/	/	/	/
Metering	x	/	x	/	/	/
Policy control	/	/	/	/	/	/
Notification service	/	/	/	/	x	x
Orchestration	x	/	x	/	x	/
Value-added services						
Availability zones	x	x	x	x	x	/
High Availability	x	x	x	x	x	x
Hybrid support	x	x	x	x	x	x
Live migration	x	x	x	x	x	x
Portability support	/	/	/	/	/	/
Image contextualization	x	/	/	x	/	/
Virtual application support	/	/	/	/	/	/

Table 3.2: IaaS survey flexibility overview.

component that performs the jobs related. OpenStack takes advantage of the Nova API to process the compute request, which is a complete option with more taks (eg., nova-scheduler, nova-network) . The nimbus has a widely explored component, the workspace that among other tasks, processes the compute jobs. Additionally, CloudStack uses the libcloud to answer the compute requests, while OpenQRM also has a unclear definition of a component for individually respond the computer tasks.

Concerning the storage on a cloud pool, several components perform related jobs. The main responsibility of this part is to offer (create, allocate, edit, delete) virtual disks to the cloud instances. Among the large storage technologies and disk formats available, the component that execute this jobs is supposed to be as flexible as possible.

Also, the last topic of the resource abstraction layer is the network. In this layer, every tool support a virtual network for the cloud due to the clear need for communication and data transfer among the virtual servers. An important detail related to high speed network is the throughput and latency which are often defined by the physical equipments. On the other hand, the network algorithms for manage the queue are often related to the cloud components, which is has different approaches on each solutions and deployment.

Furthermore, on the core service layer is shaped by several topics. One of the most important is the identity service that is important on complex cloud deployments for guarantee the authorization on the cloud manager jobs. It is highlight the OpenStack keystone API that is always on the tasks of the infrastructure management. The scheduler is important because it prepares all the management jobs according to the resources availability and load balance. Additionally, the image repository provides a catalog of pre-built OS ISOs to facilitate the instantiation process. For charging and billing, only CloudStack and OpenStack enable these services, they are largely used on public cloud providers in order to control the clients usage. All the tools support the logging act of clients on the cloud system.

The support layer is shaped by additional services on a cloud system. The

Support	OpenNebula	Eucalyptus	OpenStack
Resource abstraction layer			
Compute	Oned	Internal	Nova
Storage	Internal	Walrus	Object storage (Swift) /Block Storage (Cinder)
Volume	Internal	Storage Controller	Nova-Volume
Network	Virtual Network Manager	Internal	Neutron/Nova-network
Core service layer			
Identity service	Internal	IAM API	Keystone
Scheduling	Scheduler	Cluster Controller	Nova-scheduler
Image repository	Internal	Internal	Glance
Charging and billing	/	/	ceilometer
Logging	Internal	Internal	Internal
Support layer			
Message bus	Internal/RabbitMQ	/	RabbitMQ
Database	sqlite/MySQL	Sqlite/HSQLDB	MySQL/Galera/MariaDB/MongoDB
Transfer service	Internal	Node Controller	Nova Object store/cinder
Management layer			
Resource management	Internal	Internal	Nova
Federation management	/	/	/
Elasticity management	Auto-scaling	Elastic Load Balancing	Elastic Recheck
User/group management	Internal	Internal	Internal
SLA definition	/	/	/
Monitoring	probe/ssh/OneGate	External	External
Reporting	code reporting	/	/
Incident management	/	/	External
Power management	External	External	Blueprint driver
Lease management	External	External	External
Management tools			
CLI tools	OpenNebula CLI	Euca2ools	OpenStack (CLI)
APIs	Public cloud and Plugins	Public cloud and Plugins	Public cloud and Plugins
Dashboard	Sunstone(Admin UI, User UI)	Admin UI, User UI	Horizon(Admin UI)
Orchestrator	oneflow	/	heat
Security layer			
Authentication	Basic Auth/OpenNebula Auth/x509 Auth/LDAP	LDAP/ AWS IAM	LDAP/Tokens(APIs)/X.509/HTTPD
Authorization	Auth driver	Internal	Keystone
Security groups	Internal	Internal	Internal
Single sign-on	/	/	/
Security monitoring	External	External	External
Control layer			
SLA enforcement	/	/	/
SLA monitoring	/	/	/
Metering	External	/	ceilometer
Policy control	/	/	/
Notification service	/	/	/
Orchestration	OneFlow	/	heat
Value-added services			
Availability zones	Internal	Internal	Internal
High Availability	External	External	External
Hybrid support	Amazon EC2/Microsoft Azure/IBM	Amazon AWS	HP Helion/Amazon EC2/IBM
Live migration	Internal	Internal	Internal
Portability support	/	/	/
Image contextualization	one-context	/	/
Virtual application support	/	/	/

Table 3.3: IaaS survey tools flexibility(a).

Support	Nimbus	CloudStack	OpenQRM
Resource abstraction layer			
Compute	workspace	Libcloud	Internal
Storage	Cumulus	Internal	Internal
Volume	Internal	Internal	Internal
Network	workspace Control	Internal	Internal
Core service layer			
Identity service	/	IAM plugin	/
Scheduling	Internal	Internal	Internal
Image repository	Internal	Internal	Image Shelf
Charging and billing	/	CloudStack Usage	/
Logging	Workspace-service	Internal	Internal
Support layer			
Message bus	/	internal/RabbitMQ	/
Database	Internal	MySQL	Postgres/MySQL
Transfer service	Workspace-service	Internal	Internal
Management layer			
Resource management	Internal	Internal	Internal
Federation management	/	/	/
Elasticity management	/	Elastic Load Balancing	/
User/group management	Internal	Internal	Internal
SLA definition	/	/	/
Monitoring	External	External	External
Reporting	/	/	/
Incident management	/	Internal	Internal
Power management	External	External	External
Lease management	External	External	External
Management tools			
CLI tools	Internal	cloudmonkey	Internal
APIs	Public cloud and Plugins	Public cloud and Plugins	Public cloud and Plugins
Dashboard	/	Admin UI	Admin UI
Orchestrator	/	cloudstack Cookbook	/
Security layer			
Authentication	X.509	SAML/LDAP	LDAP
Authorization	Internal	SAML	Internal
Security groups	Internal	Internal	/
Single sign-on	/	External	/
Security monitoring	External	External	External
Control layer			
SLA enforcement	/	/	/
SLA monitoring	/	/	/
Metering	/	/	/
Policy control	/	/	/
Notification service	/	Internal	Internal
Orchestration	/	cloudstack Cookbook	/
Value-added services			
Availability zones	EC2 driver	Internal	/
High Availability	External	External	External
Hybrid support	Amazon EC2	Amazon EC2	Amazon AWS
Live migration	Internal	Internal	Internal
Portability support	/	/	/
Image contextualization	Nimbus context	/	/
Virtual application support	/	/	/

Table 3.4: IaaS survey tools flexibility(b).

Message bus is used for communicate among the independent services and APIs on a cloud environment. One interesting approach explored is the RabbitMQ for transfering messages among the independent components. Another service is the database used in order to store users login credentials and logs.

When looking to an IaaS cloud, probably the most important layer is the management because there all the jobs are scheduled and processsed, and users and resources are controlled on a centralized way. The monitoring is another important part of the management, which checks the hosts and instances status. In this layer, OpenNebula, OpenStack and CloudStack are more comprehensive on the overall coverage. Such relevance is important for the cloud tools.

Additionally, in the management tools layer are the interfaces for control the cloud infrastructure either for administrators as for end users. The CLIs are used in order to offer a command line prompt for a faster and advanced management. Each cloud solution presents this interface on a different way, less or more fragmented. The APIs are used to offer additional cloud components and services to users. The public APIs may integrated with every cloud, and the cloud solutions offers support to several APIs. Regarding the dashboard, the Nimbus is the only solution that does not offer a web interface to manage the cloud system (hosts, instances, volumes, users, groups, among others). Also, the orchestrator is a usefull tool for cloud deployments, and OpenStack, OpenNebula and CloudStack offers this service.

Security is one of the biggest challenges for cloud technologies. As a consequence, the IaaS solutions offer a level of security. Despite the general poor security monitoring and single sign-on (only CloudStack supports), the tools enable authentication, authorization and security groups in order to increase the privacy and isolation.

When looking to the control layer of the IaaS tools is clear that the coverage is weak. The overall SLA support is poor due to the fact that the solution are most suitable for private cloud and does not focus on such topics related to public cloud deployment. Also, CloudStack and OpenQRM have the advantage of offering the notification service, which facilitates the management and usability.

The last layer is the value-added that covers additional services to make the difference on a cloud system. The availability zones offer an isolation among the virtual data centers and only OpenQRM is not able to provide this service. The high availability is another complex and important role on cloud. One way for walk in the direction of the high availability is offer redundancy on every level, disks (eg., RAID arrangement), services and database (replicate on different services) and at least two management servers.

In an overall approach, the open source IaaS cloud tools present a relevant and advanced options either for cloud administrator as for cloud users. The OpenStack cloud system can be highlighted as the most complete solution due to the fact of its high customization and fragmentation. On the other hand, in a simplest cloud comparison, some solutions (eg., OpenNebula) may be highlighted, while OpenStack elevates the complexities for deploy a cloud virtual pool.

As presented on Table 3.3, Oned is the daemon in charge of the OpenNebula computer management. In the OpenNebula official documentation, OpenNebula (2015a) gives an overview of the oned: "The OpenNebula daemon oned manages the cluster nodes, virtual networks, virtual machines, users, groups and storage datastores". In other words, the Oned controls the computational resources (processor, memory) and delivers services to users.

Also, the OpenNebula resource for virtual machines is managed by the Libvirt (2015), which works as Virtual Machine Manager (VMM). Furthermore, the Opennebula (2015b) uses a scheduler in order to allocate the virtual machines in the available hosts, following rules of resources and workloads distribution. The most common way to perform the storage when using OpenNebula cloud is the Network File System (NFS), which is well described by Roebuck (2012). Finally, OpenNebula promises to conciliate among simplicity, openness, reliability and flexibility in order to facilitate corporation to met the demands. An important and almost exclusive OpenNebula feature is the VM context, which facilitates the VMs consolidation and deployment.

Eucalyptus presents a suitable option for management of the virtual infrastructure due to the fact of its strong and flexible capacities. Such solution has some unique features, for instance, the walrus storage (EUCALYPTUS, 2015b), which is unique on IaaS tools and is probably a welcome idea to benchmarks the I/O rates in such system. Also, it is interesting the way that the deployment is recommended to be performed and probably is a good idea to analyse it carefully in the future. Also, as concluded by Kumar and Gupta (2014), Eucalyptus is a robust and optimized solution for build a cloud environment.

Openstack offers a complete and flexible solution for IaaS cloud. The several OpenStack APIs (OPENSTACK, 2015c) communicates among each other using the RabbitMQ (2015) message broker. Such messenger runs as a interface between APIs and offers infrastructure services for the cloud system meet the demands. Among the OpenStack infrastructure resource, there are near to forty individual components available to be enabled for a cloud environment. The most unique and important to highlight OpenStack solution approach are, the orchestrator, ceilometer (metering), swift (object storage support) and hybrid support. As a consequence, OpenStack is a granular and robust solution for built complex and reliable virtual pool.

Another cloud solution, Nimbus (2015) presents the concept of a flexible and scientific cloud system with unique characteristics and features. The nimbus cloud infrastructure is fully compatible with Amazon EC2/S3 cloud in case of the need for a Hybrid cloud. However, the Nimbus toolkit allows the deployment and provision of Infrastructure as a service converting a cluster hardware into a cloud. This cloud solution takes advantage of the Cumulus storage system (HWANG; DONGARRA; FOX, 2013) through a pretty interesting features. As a result, the storage efficiency impact directly on the applications performance and probably would be great to perform simulation experiments on top of the Cumulus storage.

CloudStack is other interesting framework for IaaS cloud deployment. Such solution offers flexibility and elasticity based on the goal of support several technologies (Hypervisors, networks, storage, Database, Hybrid Support). Also, it brings advanced recipes for the installation of customizable additional plugins in order to meet specific demands (GOASGUEN, 2014). In addition, CloudStack has its own cloud CLI (cloudmonkey) and supports utilization metering and monitoring of the resources. As emphasized by Kumar et al. (2014), the CloudStack presents some original features, secure single sign on, dynamic workload management, alerts, events and notifications (CLOUDSTACK, 2015b).

Finally, OpenQRM is another open source cloud solution for data center virtualization and infrastructure as a service consolidation. The community edition of OpenQRM is free and open source. Also, one of the most interesting point about openQRM is the deployment way. The services and resources are easily configure as plugins on the graphical interface. For instance, Hypervisors, storage, network services, monitoring (Nagios), among others (OPENQRM, 2015b). Also, OpenQRM offers an interesting feature, further than the dashboard, a Cloud Portal is offered too by OpenQRM toolkit.

The article of Sotomayor et al. (2009) presents OpenNebula as a VI (virtual infrastructure) manager and adds the Haizea integration. This approach is presented in order to offer lease management for the OpenNebula cloud. The main challenge addressed by this approach present the private cloud solutions as infrastructure management that needs to efficiently use hardware resources and also guarantee the on demand services combined with QoS.

The IaaS cloud deployments have already overcome several challenges. A classical example is on the networking area. As presented by the authors Azodolmolky, Wieder and Yahyapour (2013), this approach faces a lot of different and unique issues. However, the overall network solutions improved substantially and the cloud technologies can take easily advantage of this network efficient solutions (SDN, VLAN, Infiniband, among others) . In addition, the authors Tsugawa, Matsunaga and Fortes (2014) also explored the networking challenges looking for high speed technologies to supply the cloud demand. Such high dependency for efficient network is due to clouds and inter-clouds environments transfer a huge amount of data between virtual and physical devices.

Another challenge for deploy a cloud using an open source tool is that sometimes the solution does not address the monitoring and management need. However, it is not that hard to enable an external monitoring tool in order to increase the cloud control and availability. Accordingly, the paper of Fatema et al. (2014) points out monitoring solutions for a cloud environment, which may be helpful for the decision-making. Furthermore, a study performed by Ward and Barker (2014) analyzed the monitoring solutions and cloud solutions through a specific monitoring taxonomy.

When considering the number of itens supported by the IaaS tool concerning flexbility, Figure 3.11 shows the number of itens covered by each cloud solution. Additionally, looking for a most fair approach, Figure 3.12 presents the IaaS tools support related to each layer of the taxonomy. Such results are achieved over a percentual (0% - 100%) analysis related to each taxonomy layer.

3.2.6.2 Resilience

The resilience of a computer system is given by the support for multiple heterogeneous environment and technologies to facilitate the QoS (load balance, high performance, fault tolerance, among others). Additionally, the authors of Castano and Schagaev (2015) presented resilience as the possibility of constant changes on the system.

Table 3.5 shows the IaaS tools support for enable resilience on a cloud system. The Full and Bare Metal virtualization refers to the virtualization support for Hypervi-

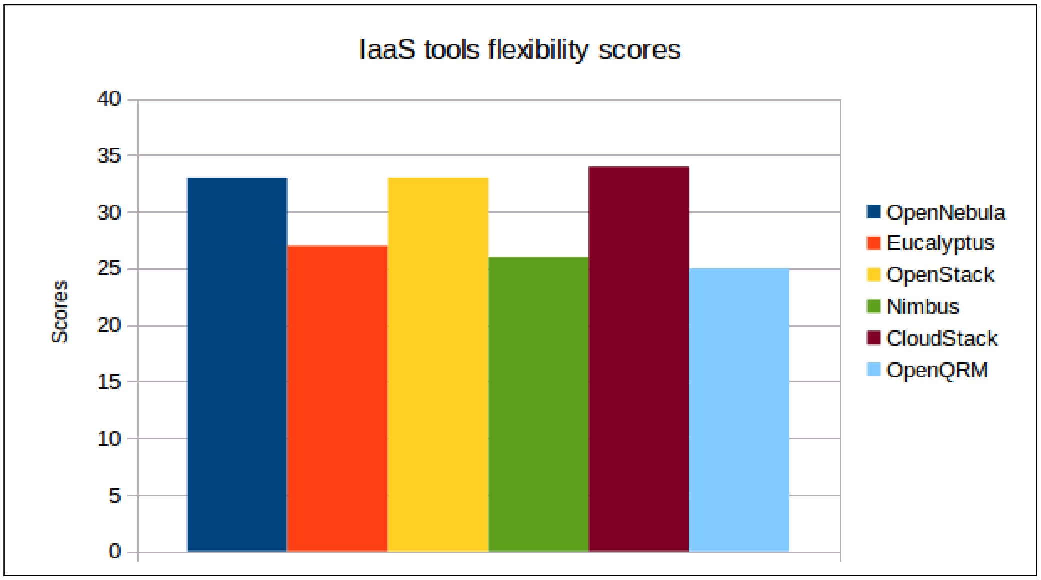


Figure 3.11: IaaS survey flexibility scores overview.

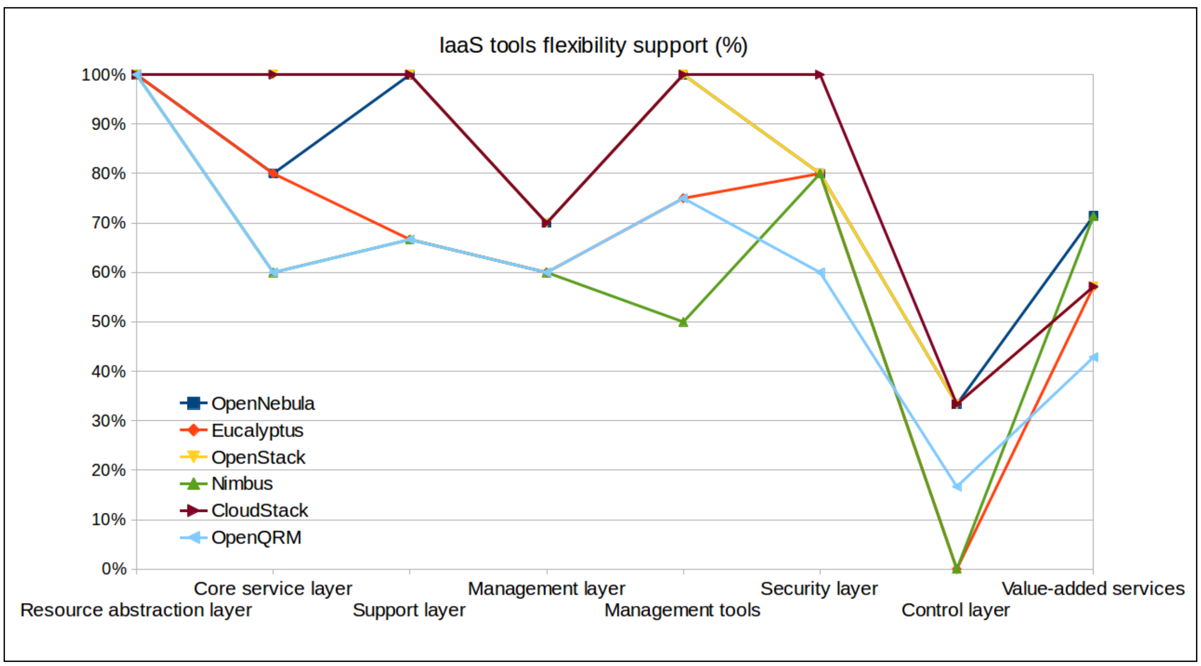


Figure 3.12: IaaS survey flexibility support percentage.

sors, Full and bare metal are significant different approaches when looking for the features and way that each one works. However, the both virtualization technologies (full and bare metal) shares the Hypervisor usage which emulates hardware resources to the virtual machines.

When the cloud storage is considered, many technologies can be used. However, depending on the cloud framework, hardware features and which virtualization technology is being used, several technologies are not possible to enable due to incompatibility issues. The topic on Table 3.5 refers to how many solutions may be used on each cloud tool. It is important on complex cloud deployments (eg., data center) due to the performance variation among storage technologies and the deployment architecture (local or distributed). Another important aspect related is the disk formats, where the compatibility vary among hypervisors and workloads support. Also, the disk formats may impact on performance due to the different hypervisor supported and I/O scheduler algorithms used. As a consequence, the most indicated disk may be quite different when considering heterogeneous workloads.

In addition, the network support is important for a cloud deployment. The network compatibility is also high dependent of the Hypervisor used and different technology may impact on the applications performance. The OS support refers to the number of host OS supported on full virtualization deployments. It is important for a cloud system because the combination among OS and hypervisor can also led to different performance results, resources consumption and system resilience. Moreover, the OS is important for the container virtualization support because the container are built inside a native OS in order to simplify the virtualization layer and so it is supposed to reduce the performance overhead.

Finally, the Web UI refers to the possibility of a cloud solution be managed and controlled over a graphical interface. Such option is well accepted due to the facility compared to command line interfaces. Furthermore, the Object storage concerns the support for a robust and new way of offering data storage dealing with disks as objects (Coyne et al. (2014)).

The average of resilience total score was 10.8 among virtualization, storage, Disks, and network. Such value is used in order to point the additional resilience

Support	OpenNebula	Eucalyptus
Full and Bare Metal virtualization	Xen, KVM, Vmware	VMware, KVM, Xen
Storage technology	NFS, ssh, ceph	ceph, OSG
Disk Formats	qcow2, vmfs, ceph, lvm, fslvm, raw, dev	qcow2, LVM, raw, vmdk
Network	dummy, ebtables, VLAN, OVS, vmware	VLAN, bridge, DHCP
OS	Ubuntu, Debian, RedHat, SUSE, CentOS	CentOS, RHEL
Container Virtualization	No	No
Web UI integration	Yes	Yes
Object Storage	Yes	Yes
Support	**OpenStack**	**Nimbus**
Full and Bare Metal virtualization	Hyper-V, VMware, Xen, KVM, VirtualBox	Xen, KVM
Storage technology	LVM, Ceph, Gluster, NFS, ZFS, Sheepdog	local, EC3
Disk Formats	LVM, qcow2, raw, vhd, vmdk, vdi	qcow2
Network	Neutron, and B.Switch, Brocade, OVS, NSX, PLUMgrid	DHCP, ebtables
OS	Debian, Ubuntu, RHEL, CentOS, Fedora, Suse	Debian, Ubuntu, RedHat, Gentoo, SUSE
Container Virtualization	Yes	No
Web UI integration	Yes	No
Object Storage	Yes	No
Support	**CloudStack**	**OpenQRM**
Full and Bare Metal virtualization	Hyper-V, Xen, KVM, VMware, VirtualBox	KVM
Storage technology	NFS, SMB, SolidFire, NetApp, Ceph, LVM	LVM, NFS, GlusterFS
Disk Formats	LVM, VMDK, VHD, qcow2,	raw, qcow2
Network	bridge, VLAN, DHCP, DNS, NVP, BigSwitch, OVS	bridge, VLAN, DHCP, DNS, TFTP
OS	Debian, Ubuntu, RHEL, CentOS	Debian, Ubuntu, RHEL, CentOS, OpenSuSE
Container Virtualization	Yes	No
Web UI integration	Yes	Yes
Object Storage	Yes	No

Table 3.5: IaaS tools resilience.

aspects (container virtualization, Web UI and Object storage) on the overall sum of the scores.

Figure 3.13 presents the scores and overall rank, which compares the IaaS cloud solutions. Also, this graph shows that OpenStack and CloudStack have lead on the overall resilience results. Such results are find due to the deep contrasts among all the cloud solutions, which is actually something normal because of the market different demands and each tool has unique goals and solution to offer. However, the decision-making act is always being harder because there are several solutions available and it needs a careful analysis. As a consequence, the cloud tools need to support as many configurations and features as possible and the most resilient tools can easier meet any demand.

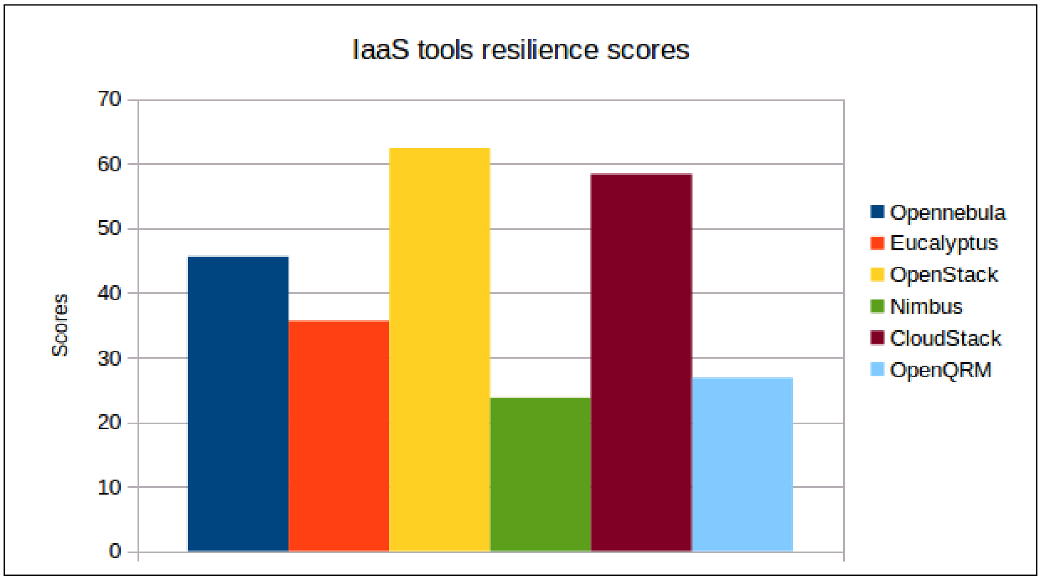


Figure 3.13: IaaS survey resilience overall scores.

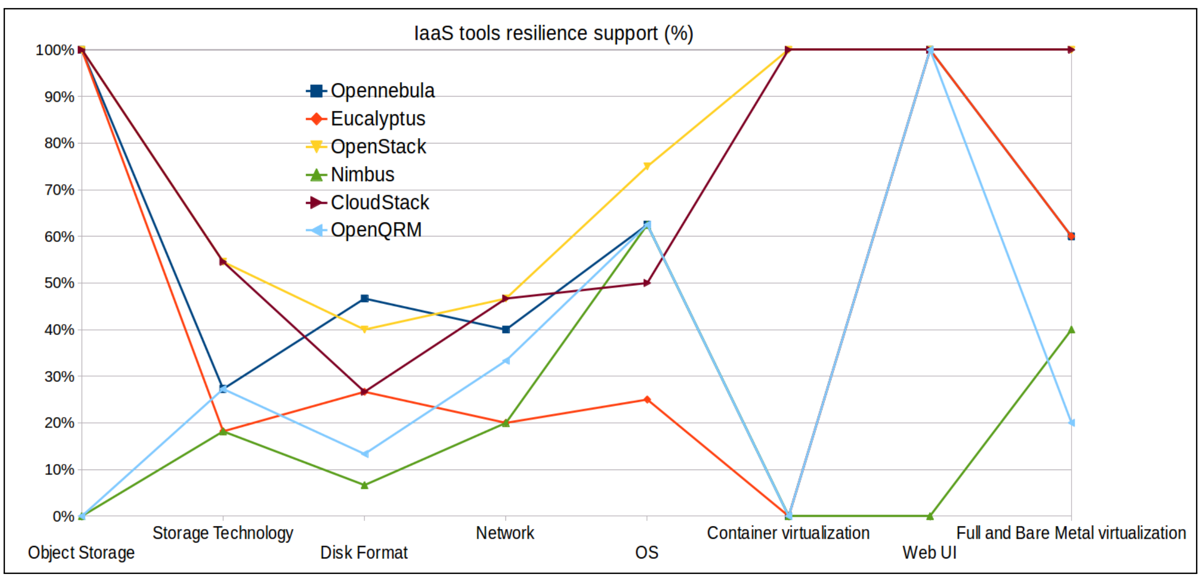


Figure 3.14: IaaS survey resilience percentage.

Furthermore, as emphasized on Figure 3.14 the cloud solutions have contrasting results. In an overall approach, OpenStack presents the best percentage of resilience, followed by CloudStack. This two solutions received the best averages mainly because of the large support for storage, host OS and Hypervisors. Also, OpenStack and CloudStack support the container virtualization (OS level). The solutions supported on both solutions is LXC (Linux Containers), which may be pretty useful on specific demands for reduce the performance overhead. Finally, the fact of support object storage is highly recommended for complex storage systems and treat huge amount of data (eg., data mining, big data).

The support for object storage is a differential, the technologies supported in the tools that enables object storage are Swift and Ceph. Also, the network plays an important role on cloud systems due to the fact of the horizontal growing of the computational infrastructure. Such deployment relies on a reliable and high-speed network in order to distribute the workload (parallelism, cluster, grid).

3.2.7 IaaS Tools Comparison Remarks

As presented on the Tables 3.2, 3.5, every tool has different features concerning the robustness (flexibility and resilience). Such factors impact directly in the capacity of the cloud system to meet the users and applications demands. Also, the most robust tools are the solution which better support the interoperability on the combination of flexibility and resilience. This ability affects the overall cloud performance, as presented on Section 3.3.

In addition, when deploying a cloud environment is easy to run into installation issues. Such events occur due to the complexity of the tools. The most granular and fragmented solution often are more difficult to install when comparing with centralized solutions. The complexity is the price of the robustness, but if a customize, highly available and satisfactory performance is needed, probably the most robust solution is more suitable for build a cloud virtual pool.

Thus, as presented on Figure 3.15, OpenStack and CloudStack have the best averages concerning robustness. This graph highlights the contrast between the solutions. For instance, Nimbus had poor robustness average is suitable and focused on cloud deployments for the scientific field. Also, OpenQRM community edition has

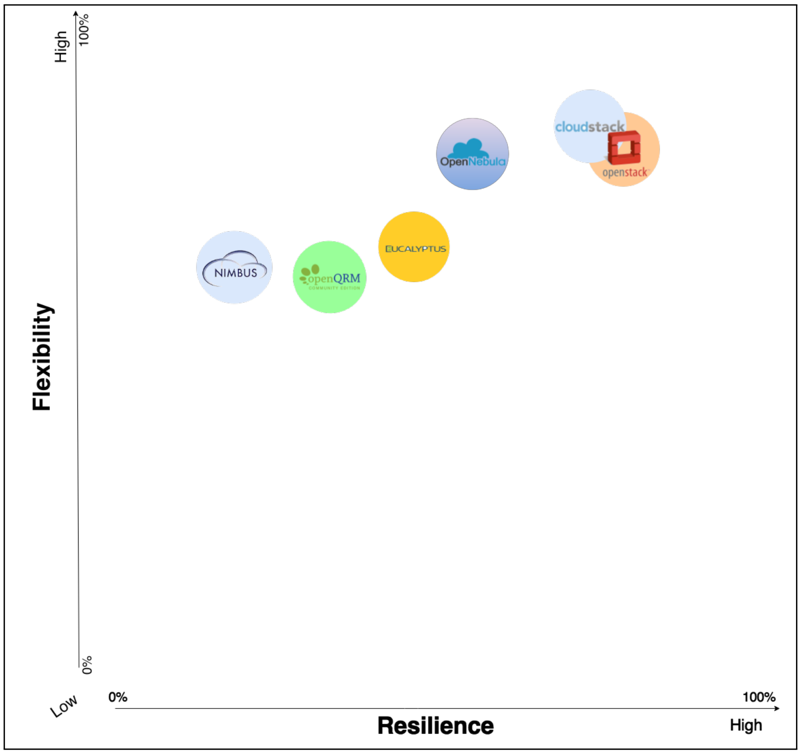


Figure 3.15: IaaS tools robustness.

limited robustness, but the installation is as easier as possible, within minutes a cloud system can be available. However, Nimbus and OpenQRM probably are weak for a scalable and complex deployment.

Over IaaS tools robustness survey is possible to clarify what tools covers and they do not. The decision-making is quite difficult due to the fact that several solutions are available and unfortunately often there is lack of information of the tools. In order to face the performance issues of cloud systems, the next section gives a special attention to resource allocation and virtualization overhead. However, there are some points that the tools can spoil the overall performance, mainly when the recommended deployment is not followed or when the tools do not support application needs. For this reason, is important the robustness of the tools, because it enables a scalable and changeable environment for clients run their application in a recommended virtual pool. Additionally, it is important to give the necessary care to the tools details because it can impact on the performance results. For instance, the OpenNebula and

OpenStack scheduler allocates the virtual machines on the server that has less load and is actually compatible with the demands needed.

3.3 SURVEYING PERFORMANCE'S STATE OF THE ART FOR OPEN SOURCE IAAS PRIVATE CLOUD

Cloud computing emerged as an alternative for running high-performance workloads. However, there are several challenges (eg., performance, security, isolation, resources sharing) related to such deployment. Additionally, almost every challenge is due to the additional abstraction layers needed on virtualized systems. Despite such difficulties, the cloud could be a highly accepted solution if the challenges are overcome and also, may offer advantages (flexibility, elasticity (Ali-Eldin, Tordsson and Elmroth (2012)), best resource usage, load balance (Xu, Pang and Fu (2013)), fault tolerance (Egwutuoha et al. (2012)), live migration (Ye et al. (2011)), among others).

In addition, the cloud offers additional flexibility either on public as for private cloud deployments. However, such flexibility increases the portfolio of options and causes uncertainty knowledge of the application performance on cloud due to several potential variations and overheads. For this reason, the sections below explore and highlight the open challenges for performance on cloud. Also, known solutions to overcome or reduce the issues are presented.

3.3.1 Survey Methodology

In this paper, a methodology has been used to select papers for the literature review and also researches that answers some of the research questions. The paper have been found on digital libraries and surveyed to get the responses required. Also, a search for keywords (cloud performance overhead) was used in order to find relevant papers followed by a a reading and analysis of each paper approach.

3.3.2 Virtualization Overhead

The virtualization is an additional abstraction layer deployed to emulate operating systems and drivers in order to achieve flexibility and best resources usage. In 2007, one of the first approach concerning virtualization performance, the paper of Willmann et al. (2007) presented the idea that a network of a VM running on top of a hypervisor can just achieve 30% of the native machines rate. Also, the same paper explained virtual networks : "the network interface and the hypervisor collaborate to provide the abstraction that each guest operating system is connected directly to its own network interface. This eliminates many of the overheads of network virtualization". Currently, the virtual networks are able to have better performance. Moreover, such technology achieved high efficiency and increased the overall performance without increase the CPU usage.

There is a small overhead on virtualized system compared to the native environment when considering the memory and CPU results. However, the obstacle becomes bigger when is considered the I/O Santos et al. (2008). Additionally, the self-virtualization: "hypervisor-level abstraction designed to encapsulate the virtualization of I/O devices" (Raj et al. (2006)). Such abstraction often is a virtual network interface or a virtual disk. The system performance based on self-virtualized achieved higher performance compared to the classical virtualization approach.

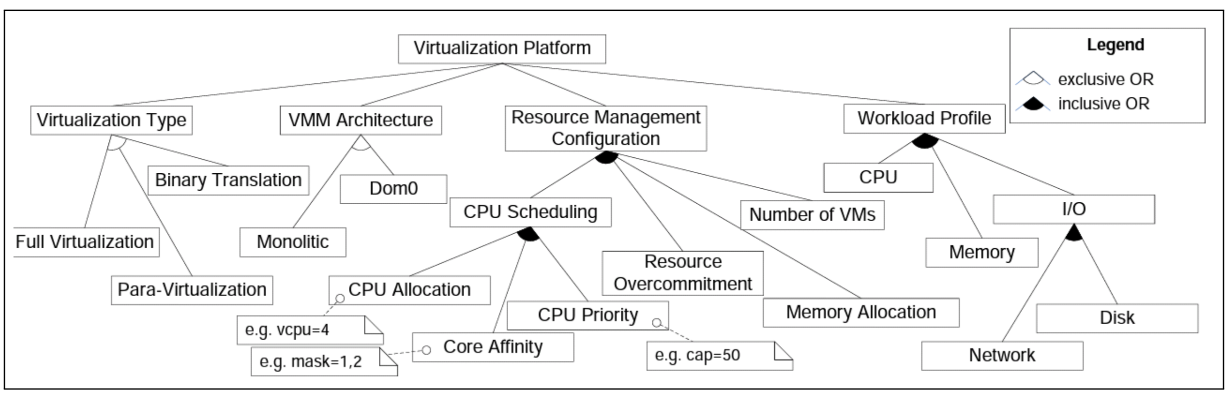
The performance degradation worst case occurs when multiple VMs are running on the same server. This deployment occurs because the processors are shared as well as the cache levels. As a consequence, the instances share the cache and one affect others Nathuji, Kansal and Ghaffarkhah (2010). Also, currently there is no efficient way to segregate the cache sizes among the VMs, such challenge for cloud provider may become a hard issue when dealing with clients SLAs and QoS.

Considering the virtualization performance issues is possible to conclude that is necessary to consider it as a whole system. In fact, virtualization impacts on the performance. However, as Younge et al. (2011) concluded, the usage of virtual system is much more than this, the advantages are many and can easily equalize the eventual degradation.

The study of Xu et al. (2014) presents a way to calculate the performance overhead between an IaaS virtualized environment (cloud) and dedicated environment that can be a native environment or a virtual based on a single VM per server (no resources concurrency among VMs). Finally, the P_v return the performance degradation.

$$P_v = \frac{cloud - reserved}{reserved}. \quad (3.1)$$

Furthermore, the study of Huber et al. (2011) analyzed the main factor which impacts on the VM by the virtualization usage, and often are the causes of the overheads.



Source: Huber et al. (2011)

Figure 3.16: Virtualization performance overhead factors.

3.3.3 Performance Variations and Standard Deviation on Cloud Systems

The authors Armbrust et al. (2010) defined the performance of applications on cloud as unpredictable. It was motivated mainly by the oscillation on I/O rates between concurrent VMs, and the CPU and memory variations. Moreover, the authors emphasized that the CPU degradation are due to the inefficient scheduling processes done by the Operating System. Also, the network I/O variation is a high concern presented on the paper of Wang and Ng (2010).

In addition, a research concluded that the performance results, impact and variation depends on which kind of applications have been running in the cloud Iosup, Yigitbasi and Epema (2011a). This happens because each workload is unique, the programs, and the deployment and architecture way is always different. Also, the operational system and the hardware interactions are performed, and finalized under several impacts. Moreover, the study of Barker and Shenoy (2010) concluded that Amazon EC2 instances have significant disks I/O variation.

Another important point is that VMs have more computational power available do not always present the best performance results when comparing with smaller VMs Hajjat et al. (2015). The way that the OS schedules the process and threads impacts on the response time for the application requests be processed. As a consequence, when multiple applications are running, the response time trend to be higher. On the other hand, on smaller VMs often a reduced number of application can run, which results on less concurrency and theoretically it can achieve better performance results for some workloads.

Also, the research of Pu et al. (2010) accomplished an approach looking for parallel application interference using the Xen Hypervisor. The same paper concluded that intensive workload may cause overhead when running on shared hardware. Such degradation are caused by the high number of context switches on the processors controlled by the virtual machine manager. Another important performance aspect found by the authors was the called "CPU contention" which happens when high intensive workloads needs an increasing I/O rates and the interference in the I/O channels. Additionally, another important aspect of the performance variations among identical instances on public cloud is analyzed by Cerotti et al. (2012). Concerning the cloud provider instances, the authors concluded that the main source of variation was the contrast among the different CPU models used on the same instances flavors.

Additionally, the variability and standard deviation issues on cloud occur on several aspects. The study of Leitner and Cito (2014) and El-Khamra et al. (2010) presents an interesting benchmark approach on public cloud providers. However, the most inquisitive presented are the variation concerns. The paper analyzed several aspects of cloud workloads fluctuation. The results suggest that the cloud is susceptible for a diversity of variations, starting in the hardware and network of the providers. Also, even on identical instances, the variation may occur due to the virtualization fluctuation. Among the resources (CPU, memory and Disks I/O), the variation is different and unpredictable. Finally, different applications and workloads show variation in the results even when running on the same instances. On the other hand, the paper of Aida et al. (2013) concluded that the fluctuation was smaller in a comparison over Hadoop on different clusters technologies.

$$P_v = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2}. \tag{3.2}$$

As shown at the equation 3.2 of Xu et al. (2014), the performance variation (P_v) is an adverse factor in the workloads performance. Also, the x_i is the VM performance result. Often, the further average is used to find the variance. Additionally, the N is how many experiments were performed. Finally, the $\bar{x}$ is related to average of the performance considering the results and number of repetitions. Additionally, the paper of Schad, Dittrich and Quiané-Ruiz (2010) presents the same equation for calculate the performance variation. However, it has received the caption of Coefficient of Variation (COV).

Moreover, using such variation equation, several researches analyzed the variation on cloud systems. The study of Tudoran et al. (2012) compared the servers response time for scientific applications. In this approach, a deep comparison among public (Azure) and private cloud (Nimbus) was done in order to explore the results differences. The conclusion presents Nimbus as a most suitable solution due to the smaller workloads variation. Additionally, the paper of Dejun, Pierre and Chi (2010) compared the response time on Amazon EC2 instances and found 8% of variation as the worst case (small size instances). Also, Table 3.6 shows an overview of the performance variation.

APPROACHES	ENVIRONMENTS	EXPERIMENTS	RESOURCES	VARIATIONS
Armbrust et al. (2010)	Amazon EC2	STREAM	Memory	4%
Armbrust et al. (2010)	Amazon EC2	Bandwidth	Disk	16%
Iosup, Yigitbasi and Epema (2011a)	AWS and Google	variation/time	Workloads	up to 40%
El-Khamra et al. (2010)	Amazon EC2	GMRES	Single CPU core	25%
Leitner and Cito (2014)	Amazon EC2	sysbench	CPU	up to 40%
Leitner and Cito (2014)	Amazon EC2	MBW	Memory	up to 40%
Leitner and Cito (2014)	Amazon EC2	sysbench	I/O	up to 27%
Leitner and Cito (2014)	Google CE	sysbench	CPU	up to 3%
Leitner and Cito (2014)	Google CE	MBW	Memory	up to 6%
Leitner and Cito (2014)	Google CE	sysbench	I/O	up to 4%
Barker and Shenoy (2010)	Amazon EC2	Jitter	Disks I/O	up to 50%
Schad, Dittrich and Quiané-Ruiz (2010)	Amazon EC2	Ubench	CPU (large instances)	24%
Schad, Dittrich and Quiané-Ruiz (2010)	Amazon EC2	Ubench	CPU (small instances)	21%
Schad, Dittrich and Quiané-Ruiz (2010)	Physical cluster	Ubench	Memory	0.3%
Schad, Dittrich and Quiané-Ruiz (2010)	Amazon EC2	Ubench	Memory (large instances)	10%
Schad, Dittrich and Quiané-Ruiz (2010)	Amazon EC2	Ubench	Memory (small instances)	8%
Schad, Dittrich and Quiané-Ruiz (2010)	Physical cluster	Bonnie++	disk I/O	0.6% to 1.9%
Schad, Dittrich and Quiané-Ruiz (2010)	Amazon EC2	Bonnie++	disk I/O	9% to 20%
Schad, Dittrich and Quiané-Ruiz (2010)	Physical cluster	Iperf	Network I/O	0.2 %
Schad, Dittrich and Quiané-Ruiz (2010)	Amazon EC2	Iperf	Network I/O	19%
Tudoran et al. (2012)	Nimbus	RPC(HTTP)	Response time	0.2%
Dejun, Pierre and Chi (2010)	Amazon EC2	Web server side	Response time	1.6% to 8%

Table 3.6: Performance variation overview.

- **Workload Variability**

  The study of Iosup, Yigitbasi and Epema (2011b) presents a performance analysis of public cloud providers. Among the performance differences, there were found significant variations either on applications and in the virtual instances. As a result, sometimes , an application may run into performance issues, while another system could easily achieve the expected rates.

In addition to that, the researches of Ghosh et al. (2013) and Scheuner et al. (2014) concluded that the results for running applications may vary sometimes. Also, it is result of the hardware, operating system and scheduler impact over the workloads.

Finally, an interesting and convenient approach is presented by the authors Alameldeen, Wood et al. (2003). This research concerns about the workloads variations and shows that often happens the "time and space variability" either on "real machine and simulation experiments" (eg., synthetic benchmark). Also, this paper presents a methodology in order to increase the precision of the results and the research points out the OS scheduler as the major responsible aspect of variations.

Despite the challenges for moving workloads to the cloud, is possible to take advantage of several cloud aspects (flexibility, elasticity, fast and easy deployment, among others) in order to improve the QoS, as presented on the researches of Wolski and Brevik (2014) and Mishra et al. (2010).

3.3.4 Public Cloud Performance

The study of Iosup, Yigitbasi and Epema (2011a) concluded that the main public cloud providers have had a poor performance for data intensive application. Such results could be found mainly due to fact of the sharing resources and virtualization penalties.

Considering the challenges related to increase the performance of application running on cloud, there are several points which needs to be taken into account. Firstly, the cloud offers elasticity without an initial investment. The key of such advantage is to rely on distributed system, which needs an efficient and reliable network. The paper of Mauch, Kunze and Hillenbrand (2013) shows an overview of high performance on cloud system, and highlights how important is a high speed network. Another important aspect of cloud services is the management, it starts over an efficient resources scheduler which allocates the demand customized among the available servers. Related to this demand, the paper of Lucas-Simarro et al. (2013) explored the cloud schedulers and design an alternative algorithm in order to facilitate the management. Also, this study analyzed the cost efficiency among public providers and concluded that sometimes there are differences related to cost and QoS.

Finally, the study of Marathe et al. (2013) compared a traditional cluster for HPC environment with the public cloud Amazon EC2 instances. Also, it presents the comparison looking for performance and cost-effectiveness. The paper emphasizes that EC2 are billed over the usage, while the native cluster is harder to estimate the cost because the infrastructure needs to be purchased previously and has further management outgoing. In contrast, the study of Roloff et al. (2012) found dissimilar results. This paper also performed the comparison among public cloud providers and native clusters. However, concluded that cloud is the best option for high performance (27% better performance and 41% cost effective).

The literature presents according to what the scientific field is being working on, and emphasizes that performance is still under overhead when compared with the native clusters. However, when looking into cost-effectiveness it may change, due to the fact that cloud is significant inexpensive. Moreover, it is clear that the performance will always depend on the applications and which are the computational power expected.

3.3.5 Private Cloud Performance

As presented on the literature review, a private cloud is a kind of deployment inside the companies domain. In such cases, it is used the enterprise hardware and the IT division is responsible for manage and control the cloud system. Several open source solutions are available for build a private cloud, and the decision-making process needs to consider a general view regarding the company needs (demand) and the solutions available. In the Section 3.2, the open source solutions are surveyed regarding the support for robustness and the most suitable deployments.

Regarding the performance challenges, on private cloud the issues are still almost the same as on public cloud, the main differences are the way of the services provisioning. Even on private cloud, the Hypervisor continue playing an important role, as presented on the study of Reddy and Rajamani (2014), which analyses three virtualization (XenServer, ESXi and KVM) solutions, run experiments and compares concerning the resources and the parallel applications performance. Also, the cloud environment was powered by CloudStack cloud management. The results highlight the deep differences among the virtualization solutions, and no one was better at all. As consequence, the hypervisor must be chosen concerning the suitability related to the overall platform features.

In contrast with the previous studies which focused on analysis of the hypervisor impact. The research of Maron et al. (2015) presents a comparison among two traditional open source cloud solutions (OpenStack and OpenNebula). The study used synthetic benchmarks, NAS MPI and OMP (Bailey et al. (1991)) in order to simulate workloads. The results demonstrated the OpenNebula as a more efficient solution, which presents best averages on compute and network resources.

Considering the advantages that a private cloud brings and the improvements offered by the open source solutions is possible to expect fair cloud applications results.

Cloud as a solution, it is a fact that they need to still the improvement process, and for the future present interesting approaches and QoS improvements.

3.3.6 Related Works

In this section the related papers are presented. Such researches have been published during the recent years and they presents interesting approaches for analyze the performance on cloud.

The study authored by Sun et al. (2014) aimed to introduce an approach of metrics to measure and compare the cloud performance by users (workload consumers and life cycles). Such metric seeks for represent the real cloud user demands. Additionally, this paper presents a case study concerning elasticity, deployment and security of a private cloud. The authors used benchmarks in order to compare the clouds performance, such as Tradelite (CPU and database), SOA Bench (Network I/O and CPU) and TPC-E (Transaction Processing workload).

The study performed by AL-Mukhtar and Mardan (2014) shows up a convenient comparison between private cloud solutions (CloudStack, Eucalyptus). Such comparison concerns the performance of the infrastructure resources isolation and the virtual pool management process. Also, this paper presents experiments concerning the performance of web applications running on clouds. The experiments were conducted based on identical hardware (3 servers each experiment) with the same host OS (CentOS 6.3), benchmarks were deployed. The infrastructure benchmarks selected were Linpack and Lookbusy (processor), Bonnie++ (Disks) STREAM (memory), Iperf (Network throughput) and the overall system was tested using UnixBench.

The results of AL-Mukhtar and Mardan (2014) emphasize the high impact of the disk deployment on VMs performance. The authors found out CloudStack NFS disk 69% faster for VMs Deployment compared to Eucalyptus local Disks. However, the Eucalyptus I/O disks had better performance than CloudStack. Additionally, the web hosting performance showed CloudStack as the most indicated by its stability and overall good performance (infrastructure isolation) .

This study of the authors Cerier et al. (2011) presents a comparison of two IaaS open source cloud solutions (Eucalyptus, Xen Cloud Platform). This paper aims to provide reference of the process of build cloud for business or for the scientific field. The results shown a possibility of use both tools to deploy an efficient cloud for users. Furthermore, the authors highlighted the different issues faced to install both clouds and the fact of the implementation be entirely different.

The paper of Silva et al. (2013) experiments cloud environments using CloudBench framework. Such benchmark is an open source suite implemented in Python which simulates application intending to experiment the system. This paper presents experiments in cloud providers. The focus was on performance, high availability and resource isolation. Thus, the authors emphasized how convenient and practice is the usage of ClodBench due to the advantages (easy management, friendly interface, monitoring, among others)

The paper presented by Gillam et al. (2013) presents a comparison and benchmarks of IaaS cloud providers (AWS, Rackspace, IBM, OpenStack). Also, the paper intents to improve the way of the execution of the experiments looking for precision and fairness. The benchmarks chosen are separated based on computational components, STREAM (memory), Linpack (CPU), Bonnie++ and IOzone (Disk I/O), and Bzip2 for stress several components through compression demands. Also the network throughput and latency was experimented (Iperf, MPPTEST and simple speed test).

Contrasting with the related papers. In this paper is proposed a complete view for performance on cloud. Obviously, run experiments on cloud is important for know the performance. However, have knowledge about the aspects impacting on the performance and fully understand the results of the samples is needed too. As a consequence, in this paper is expected to point out the factors of performance overhead and variation, and present alternatives in order to reduce the performance losses on cloud. Additionally, the reliability rate of workloads is analyzed aiming at improve the way of the definition of the confidentiality.

3.3.7 A Closer Look for Performance in Cloud

The performance results from the literature combined with an analysis suggests as cloud a fair solution for run HPC applications. However, such combination is still needing improvements and will bring interesting solution during the next years. In order to address the main alternatives to increase the performance and QoS for clients, below, some quite accurate approaches are presented.

Regarding important factors on cloud management and performance, the scheduler is one of the highest. In that way, the paper of Sun et al. (2011) introduced an alternative for improvement on the scheduler and queue management. Such approaches aims to achieve a best resources usage over an optimized VM deployment and host selection. It is considered an strategic point due to the fact of the flexibility brought by cloud services and the demand for best hardware usage QoS.

Moreover, when there is a need for test and check how the cloud environment act over high demand workloads. The study of Iosup, Prodan and Epema (2014) introduces an approach for benchmark and analysis cloud, over a complete statistical methodology for data analysis. This research may be an important helper for cloud administrator and IT manager in order to simulate their cloud capabilities.

In addition of the IaaS cloud layer, there is above and upper layers (eg.,cloud management) in a cloud environment. Such abstraction are presented on the study of Celesti et al. (2013). In this approach is possible to understand how cloud is already suitable for run business workloads, and will become even more important.

One of the first research which has ran benchmarks on cloud concluded that the cloud is suitable for host high performance applications (HE; ZHOU; KOBLER; DUFFY; MCGLYNN, 2010). Also, this paper realized about the performance overhead (little) and that the major public cloud providers were not focusing on high performance yet (looking only for business applications).

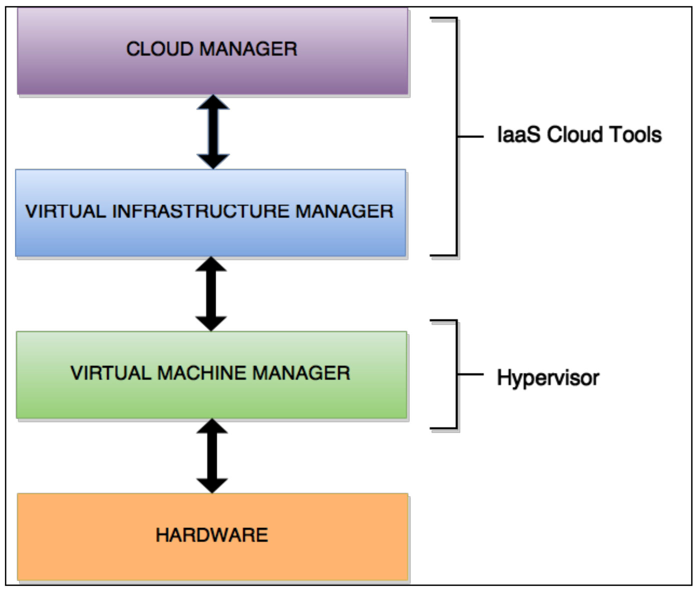


Figure 3.17: IaaS overview. Adapted from Celesti et al. (2013).

As presented on Figure 3.17, the cloud architecture involves different layers. Such architecture can be seen as a stack, where in the bottom are the physical resources (hardware). Above, there is the Virtual Machine Manager (VMM) that supports and provides the abstraction needed for the upper layers. The VMM is often an Hypervisor running within the native Operating System (Full virtualization) or a Hypervisor running direct in the hardware (Bare metal). Moreover, the Virtual Infrastructure Manager (VIM) is often the IaaS cloud tools, which takes advantage of the virtual resources in order to offer services to clients. The main tasks (resources scheduler, images, networks, volumes, templates and VMs creation) performed by the VIM are related to virtual instances and appliances (XU; LIU; JIN; VASILAKOS, 2014). Finally, the cloud manager can be a part of an IaaS tool or sometimes is done by a separate layer. The liability of such layer is to control the usage (users, groups, permissions, quotas, etc.) and QoS issues of the cloud environment.

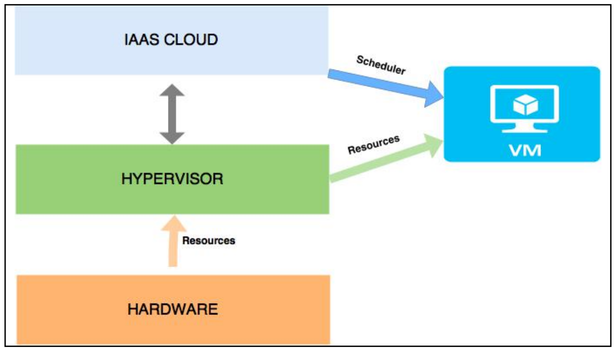


Figure 3.18: IaaS resources provision.

Figure 3.18 shows a basic virtual pool built in on a server. Cloud computing combines different technologies to offer services over the internet. The hardware resources are virtualized using a Hypervisor motivated by the advantages of such system. The first cloud layer is the IaaS, which is responsible for schedule the resources and control the utilization. On the other hand, the Hypervisor receives the demand from the cloud and actually provides such computational power. In such system, the cloud tools controls the virtual pool in order to make available the virtual resources as services. In other words, the cloud schedules the resources and transfer the demand to the Hypervisor, which offers the computational power for running the cloud instances. As a consequence, the performance issues are related to the virtualization layer (ANTONOPOULOS; GILLAM, 2010). Also, the IaaS tools are not supposed to affect the performance of virtual machines when the VMs are allocated on the suitable server and properly configured.

3.3.7.1 Virtualization Improvements

Despite the many challenges (variations and overheads) related to cloud performance, the paper of Zhu and Tung (2012) presents metrics for cloud providers maintain the QoS levels. As expressed by the authors: “A critical requirement for effective consolidation is to be able to predict the impact of application performance”. The ability of plan the resource consumption and eventual interference among VMs is considered key-issues.

A practical approach done by Novakovic et al. (2013) presents an efficient way to solve the interference issues related to multiple VMs running on the same server. This research presents DeepDive, a system for control the interference among VMs, and when the issue is true, the VM that is causing the overhead can automatically be migrated to another server. As a consequence, such model protects the stability of the overall system and appliances.

The virtual pool built on a cloud system shares the processor cache among running VMs, as a consequence, the context switches need to save the cache and transfer among cores and processors of VMs dynamically. It becomes a problem due to the fact of resources concurrency between VMs and excess of context switches. The authors of Govindan et al. (2011) emphasized such issue presenting a software to measure the VMs interference, resulting on performance overhead.

Another important aspect which needs to be considered is the virtual machine monitor (VMM) due to the importance for the virtual environment. A part of the VMM that is still demanding improvements is the I/O because of the many challenges (eg., overhead, variation) that the researchers are facing (DAI; SHI; QI; REN; WANG, 2013).

3.3.7.2 Improve the Network I/O and Disk I/O for Virtual Resources

The virtual networks largely used on cloud system are essential for the cloud services provision. Several researches were done in order to explore and im- prove the communication and transfer rates. One classical approach was introduced by Chowdhury and Boutaba (2010), which surveyed the virtual cloud solutions. Also, the paper of Huang, Carroll and Perretta (2014) presents an interesting approach concerning complex network management and high performance presenting a smart router concept.

Furthermore, regarding the I/O performance, is important to give the necessary attention to the I/O scheduler, both on host OS and on the VM side. As the paper of Boutcher and Chandra (2010) shows up an interesting point to increase the I/O rates on cloud virtual environments. Such study presents the theory that each kind of demand of application needs a specific I/O scheduler. Also, often the default Linux OS scheduler degrades the throughput.

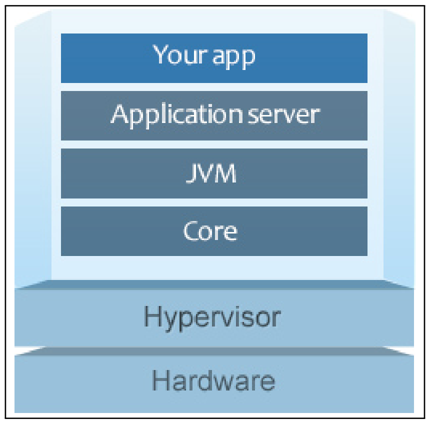
The computational applications are still increasing the demand for processing and I/O power. One important aspect to improve is the network, due to the fact that currently a lot of workload and storage system have been running distributed. In order to address the network challenges, the scientific field presents interesting solution. The research of Radhakrishnan et al. (2013) presented NicPic. NicPic is a system for improve the network scheduling in such way that reorganizes the tasks in order to increase the rates and scalability levels without increase the CPU utilization.

Related to disk I/O issues and bottlenecks, the paper of Nicolae, Riteau and Keahey (2014) presents an elastic storage cached solution. The solution is based on a framework which transparently allocates faster I/O server for cache while slower servers provides the resources. This is an interesting solution for virtual cloud environments. When the virtual disks are shared, the bottlenecks often affects the overall performance. When using faster devices for I/O cache, the throughput achieves is considerable higher.

In addition to that, concerning virtual Disks I/O is highly recommended a frequently care of the disk formats. An interesting approach is given by Tang (2011) which introduces a new disk format, Fast Virtual Disk (FVD) outperformed QEMU on several aspects regarding data I/O. Also, it shows how important is an aware configuration of the environment following the hardware, software and application way.

3.3.7.3 Cloud OSv

OSv is an open source operating system desired for run individuals cloud applications (OSV, 2015). Moreover, the main goal of this project is to increase the performance of several applications running on a cloud system. Also, the way proposed for this approach is to run a JVM (Java Virtual Machine) as an instance and only one application per instance, in order to avoid the resources isolation overhead.



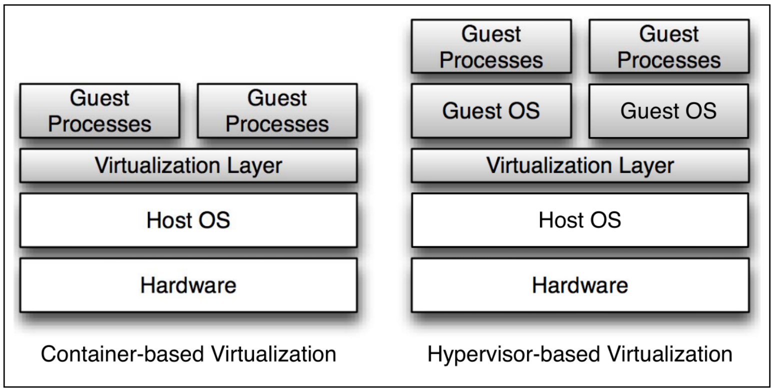
Source: OSv (2015)

Figure 3.19: OSv architecture.

Such OS run on top of a Hypervisor and can be managed by a cloud system. Also, the paper of Kivity et al. (2014) presents an academic view of this system, aiming at provide accurate information and details about it. This OS was built based on C++ programming language using a kernel and library OS coded from scratch.

3.3.7.4 Container Virtualization

The decision-making when moving to the cloud is often tricky. The previous sections describe the cloud solutions and the flexibility of the tools in order to offer advices of the most suitable option for the scenarios. However, when looking for improving the system performance further than just select powerful hardware, is really important be aware of the softwares (OS, hypervisor, plugins, drivers, programs). Since the cloud started raising, the demand for virtualization is still increasing and there is available different deployments and hardware interactions (Full Virtualization, Para Virtualization, OS virtualization).



Source: Xavier et al. (2013)

Figure 3.20: Hypervisor and container virtualization.

The OS virtualization or also known as container virtualization is a type of technology that supports multiple VMs running on the same hardware and sharing the native OS. Such approach is just possible due to the efficient isolation (users, resources, processes) used. The main solutions for containers are based on linux, OpenVZ, LXC and VServer. As a consequence, only Linux instances can run on such Hypervisors which is the price for decreasing the virtualization overhead (I/O, processing, network).

On the other hand, the containers machines loose in part the flexibility achieved by the traditional virtualization because of the restriction on compatibility of the platforms and OS. Using containers, the security concerns may become issues due to the fact of the integration and share (Hardware, OS, kernel).

Despite the challenges, the OS virtualization is a promise technology that offers strong advantages. Looking on the scientific field, the containers are under analysis, Xavier et al. (2013) presented a performance approach considering resource isolation and HPC applications, and emphasized OS virtualization as significant better using than the traditional (hardware) virtualization.

In addition, the study of Tafa et al. (2011) explored 5 virtualization solutions

(KVM, Xen-HVM, Xen, ESX-server, Open VZ) among different hypervisor implementations (Full virtualization, Para Virtualization, OS Virtualization). The paper results shown that OpenVZ (container) is more efficient and simplest that the other Hypervisors and is faster because of the kernel sharing. However, the OpenVZ guestOS isolation are still a challenge, contrasting with the several performance advantages.

3.3.7.5 A Custom Approach

Also, several aspects need to work together in order to the cloud system achieve the expected performance. For instance, the study of ExpóSito et al. (2013) concluded the cloud demand for more scalable middlewares in order to support the wished flexibility and elasticity levels. Consequently, after achieve flexibility the workloads can be distributed over a reliable network and offer performance on demand.

The study of Ashwini, Divya and Sanjay (2013) emphasizes cloud as a solution for HPC through the utilization of cluster for scalable and flexible environment for the applications. Despite the heterogeneity of such system, the authors proposed a framework in order to select the most suitable VMs and cluster for the applications. The memory management promises to be a strong point, over an efficient mapping process combined with the rule of avoid process interoperability (no context switches).

In addition to that, a cloud system is suitable for complex and huge deployment environments. Several cloud may be connected and share resources over networks (LAN, WAN). Such advanced cloud is called Inter-cloud, and the literature offers interesting points of view. The study of Toosi, Calheiros and Buyya (2014), surveyed and analyzed the solutions for efficient interoperability and availability of the services, which may achieve an powerful and efficient computational system that never has been possible before. Also, the Inter-Cloud deployment is introduced looking for improve the current QoS in the study of Grozev and Buyya (2014).

At the end of the research exploring the cloud performance challenges and potential solutions, became clear that is impossible to have a single recipe. The cloud is a complex and under improvements technology. Before a careful analysis is impossible to predict the performance of an application in the cloud. The decision for move or not the cloud needs to consider everything, no half consideration and conclusions because of the importance of such process. Early or late, such decision was or will be a milestone for every enterprise and research group.

On the other hand, now it is clear that the cloud challenges outperforms its challenges. The cloud is definitely suitable for enterprise applications and still improving day by day the results on high-performance applications. The corporations need to consider and choose the better cloud type if the public cloud, private or hybrid is recommended in order to meet the demands. Before a successful virtual environment is running, several aspects needs to be considered and the most suitable option needs to be used. Starting on the hardware, the equipments needs to be chosen according to the computational power needed.

Moreover, the hypervisor, virtualization type and cloud tool needs to be selected based on the demand for robustness and the features of the workloads and clients. Finally, the high level decisions have to take into account the storage deployment architecture (local or distributed). The storage technology and most appropriate disk format. Also, the network architecture and plugin for the best performance have to be considered. Furthermore, split the amount of memory and number of CPUs is often the easiest part. The flexibility for availability zones, users groups and cloud orchestration is a differential for cloud solutions as presented on Section 3.2.

3.3.8 A Proposal Model for Getting the Workload Reliability Maximum Rate

The reliability can be measured using several techniques. On the Engineering and Computer Science field, the reliability is defined by numerical average among the results that presents a regular test performance and others that presented impact (AGGARWAL, 2012). Also, the reliability rate is often used to test hypotheses and analyze if the experiments present or not statistical significance on the differences, as presented on Section 2.1.5.

Related with this paper, which concerns workloads reliability rates, there are several researches done for increase the reliability of the experiments on different majors (eg. Health Science, Computer Science, Chemistry, Physics, among others). In the paper of Tian, Rudraraju and Li (2004), an approach for the reliability of web servers has been presented. Additionally, the study of Guenter, Jain and Williams (2011) performed an analysis on energy distribution systems for increasing the reliability. Additionally, the paper of Shin and Win (2009) presents a reliability approach in order to increase precision of the communication on MIMO channels over a formula.

The paper of Schad, Dittrich and Quiané-Ruiz (2010) finds different variation across resources running on a native cluster. In the network, the variation between the experiments was 0.2%, while on memory and Disks I/O presented higher rates (0.3% and 0.6% to 1.9% respectively). As a consequence, each resource has its own reliability rate and it needs to be taken into account. Such variations affect the reliability rates. Therefore, is not just to assume the same reliability rate for different workloads. It becomes more critical in a scenario where you will compare same environments and test the impact of tools and workloads.

$$WRMR_s = 100 - (((\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}{}_x(x_i - \overline{x})^2}) * 100)/\overline{x}) \tag{3.3}$$

In general, it is possible to define error rates and assume the reliability rates, which means how much the experiment may be trusted. However, in this paper approach is not recommended to assume a static error rate because the comparison is among different computational aspects. Additionally, is required a reliability rate in order to answer if the IaaS cloud tools impact or not in the performance. Another problem that makes difficult the performance prediction are the hardware and the workloads variation due to the unstable results needs a careful analysis.

In this research is proposed the WRMR (Workload Reliability Maximum Rate) equation below. It is the maximum reliability level that the experiments of the workloads may achieve. It considers the standard deviation and the mean (N) and the sample (s). Also, the "x" represents the variable vector of the input while the "i" corresponds to the index of the equation. Such equation is presented in order to have an accurate way to measure the standard deviation and get the reliability rate. The standard deviation for samples act as a "correction" in order to avoid and subtract the errors rates. Further, calculate the standard deviation on workloads.

The homogeneity and isolation of the environment are required in order to find the actual impact factor of an object. In complex experiments, the hypotheses are often hard to be tested. The reliability rate provided by the WRMR is important for avoiding the influence on the hypotheses test, which can benefit one of the approaches and compromise the conclusions. Therefore, supposing that there are at least 2% of the standard deviation from a workload sample when executing in a native environment. If assumed 99% of reliability for the same workload to compare the influences of different objects in this environment, on the hypothesis test you will may have 1% influence from the native environment when observing two or more objects impact factors.

A practical example is the experiment of Maron (2014) to illustrate the applicability of the proposal model. The goal of its work was to evaluate the performance impact of the OpenStack and OpenNebula tools on isolated and parallel workloads. To do so, it was necessary to isolate two clouds with the same hardware configuration and software environment. The only differences were the tools. He also assumed 95% of reliability rate to apply the hypotheses test on all samples. Based on the proposal model, the recommendation has to be the following. An environment needs to be built without cloud tools (only hardware and hypervisor) and run the experiments to get the WRMR before perform the statistical analysis. This is because the formula gives the maximum reliability rate for a specific workload, avoiding a reliability assumption stricter than it is supposed to be. Therefore, it is possible to be sure that the influences will come only from the objects in test (OpenStack and OpenNebula).

CONCLUSION

The cloud brought the vision of the computer as a utility offering resources as services over the internet. Since the first public provider began to offer such resources, several aspects have been improved in order to provide reliable and on demand services to end users. As a consequence, IaaS become the most popular cloud product. Additionally, the open source solutions have emerged avoiding the vendors and providers lock either for private as for public cloud deployments.

In this thesis, the features of open source solution for IaaS have been surveyed looking for filling the lack of information regarding the robustness. The results shown that the tool are quite different. Several aspects present deep comparison contrasts. Related to the robustness (flexibility and resilience), OpenStack and CloudStack had the best averages, and the OpenNebula also has fair results. The major aspect was OpenStack outperforming the other solutions due to the wide resilience supported, regarding storage, network, hypervisor and OS support. On the other hand, the CloudStack overcome due to its huge flexibility for services such as support to enabling notification services, orchestrator, user utilization metering, single sign-on on the entire system, among others. Finally, the robustness is quite often affecting the performance due to the variation induced by the storage, hypervisor and host OS over the workloads and applications.

Furthermore, a cloud performance survey was conducted in order to identify in the literature the main issues and challenges related to the performance of applications running on the cloud. Consequently, the potential ways to reduce the overhead and improve the cloud QoS were pointed out. It was done aiming at being a reference and address the lack, which needs to be studied and analyzed in the future. Currently, it is still hard to compare the performance averages between the native and virtualized environment.

In addition, the virtualization goes further than just CPU and memory emulation, it covers also virtual disks and network. The interfaces (eg., drivers, APIs) between hardware and the virtual system needed to have several changes in order to make virtualization approach even possible. However, some parts continue needing improvements. For instance, the scheduler is no longer that simple as was when only one native OS was running in the hardware. Nevertheless, the challenges are many, the scheduler algorithms are being separated among computer, I/O and network in order to reduce the overhead and increase the granularity and interoperability. Additionally, the context switches issues are being minimized, as discussed on Section 3.3.

At the end of the performance survey was possible to demystify the aspects of performance on IaaS cloud environments. The cloud tools act as a resource scheduler and can impact on cloud operation performance (eg., create VM, virtual disk, virtual network, users). Therefore, after the VM is running, the processing and I/O resources are provided by the virtualization technology and only managed by the cloud system. Also, according to the tools official documentation they can not impact in the VMs performance, as emphasized on Subsection 3.3.7.

A specific workload may suffer less or more impact running on the cloud, as presented on Section 3.3. The hardware can affect and the number of virtual machines running on the same server can result on performance degradation because of the resources concurrency among them. Also, the performance of the hypervisor solution vary depending on the application, and sometimes if the flexibility is needless, for example, container virtualization may be used in order to increase the resources isolation.

Through the analysis, survey, and the performance approach is possible to apply for the cloud a similar view as presented by Wind (2011), that the open source IaaS tools have several advantages and achieved improvements during the last years, and it is possible to deploy an efficient and reliable virtual pool.

Finally, it is possible to clear define the cloud tools as technologies which offers resources as services. In this thesis, the IaaS layer was explored, in such way that the components and its respective jobs were highlighted. The virtualization is quite important for a cloud system and combines a best hardware usage with flexibility. However, virtualization and cloud are not the same technology. A cloud is much more than virtualization, it is everything as services on demand to end users.

HYPOTHESES VALIDATION

The first hypothesis says that the open source private cloud solutions for IaaS will not impact on the performance when running workloads, according to their description. Related to the documentation, this hypothesis can be validated because the tools are not supposed to impact on the performance of workloads, as presented on Section 3.2. But, the cloud operations jobs may vary according to the tools (eg., time of VM deployment, volume creation, network establishment, among others) being not true this hypothesis for these situations.

The second hypothesis proposes that the surveyed cloud tools offer the same robustness levels. This hypothesis is not true because as presented on Subsection 3.2.6 and in Figure 3.15 either the flexibility as the resilience scores and percentage are distinct on each IaaS open source cloud tool. However, it is possible to highlight OpenStack as the most resilient and CloudStack as the most flexible.

The third hypothesis emphasizes that 95% of reliability is the appropriate indicator for significant different hypotheses test between open source IaaS private cloud solutions and it is not true. It is concluded because as presented on Section3.3.8 for cloud and computational samples the hypotheses need to be validate according to the specific reliability rate of the environment and from the workloads. Also, it highlights how complex and careful is required the experiments analyze in order to have confidence on the finds.

Finally, the last hypothesis which says that OpenStack and OpenNebula impact on workloads' performance as presented by Maron (2014). This hypothesis is still under research and currently is impossible to validate with confidence. It is because a new and isolated environment is needed to find the workloads reliability rate through the WRMR and after perform the hypotheses evaluation on the cloud tools. Also, at the time of the research, unfortunately, there was no cluster available for built such environment, becoming a future work.

FUTURE WORKS

The cloud seems to be a technology which is going to conduct the IT future and the number of cloud services, models and deployments are still increasing. IaaS is the lower cloud level, which results on high dependency to the layers above (platform, software). In this way, the performance of the IaaS is even more important. In the future is wished to explore this rich world and run experiments concerning the OS, hypervisor, IaaS tools, storage and network technologies, and regarding different workloads resources demand (CPU, memory, and Network and Disks I/O). Unfortunately, in this thesis was not possible to deploy cloud pools and run workloads due to the physical infrastructure limitations.

In order to continue the aspects that were not explored in this research, it is planned to explore several related and advanced matters in the future.

In the first step, the open source solutions are aimed to be installed on an isolated pool in order to analyze carefully its services. One research can be the performance of operational and management tasks on the cloud systems. It is distinguished because the solutions have different aspects and goals. As a consequence, the time for executing management jobs on the cloud for providing resources for users may vary among the cloud solutions. Additionally, it is wished to analyze how much these cloud tasks consume, concerning computational resources and energy power.

Secondly, it is clear the heavy impact of the hypervisor on a cloud system. For this reason, it is very desirable to explore all the hypervisor available for cloud and include the different virtualization types (full, bare metal and OS level), and combining with distinct types of workloads and operating systems. Nevertheless, explore such features inside an IaaS cloud environment and compare the combinations among the IaaS tools and technologies supported. It is important because is impossible to have an identical cloud pool using distinct solutions due to compatibility and support variations (drivers, components, OS, Hypervisor, VMM, emulator, among others). However, such contrast is important for find the best combinations (eg., hardware, OS,hypervisor, cloud tool, storage, network).

Third, there is an increasing interest for exploring the cloud environment performance concerning multiple instances on the same physical server (multi-tenancy). In other words, it is clear that the resources concurrency among VMs on the same server can degrade the performance. However, there is unknown how it will behave concerning the hypervisor and OS. Also, the literature presents Cloud OSv as an alternative for improving the performance and it is wished to perform experiments using it as virtual instance.

In addition to cloud infrastructure and resources provision concerns, unless that are a production environment (where is possible to install direct the enterprise applications) is quite important introduce a coherent experiments methodology. Currently, there is available the benchmark methodology proposed by Maron (2014) which covers experiments for resources isolation and high performance applications. However, in order to increase the precision and reliability of the experiments, some different workloads are expected to be run. Such workloads are designed to run over a continuous flow, without sequential repetitions. Examples of these approaches are the YCSB[3] and CloudBench[4], and both are suitable for cloud virtualized deployments.

The Hadoop is another workload which is growing interest for running on the cloud. Several public cloud providers are currently offering hadoop-as-a-service to end users. Also, the performance overhead are affecting the results of the hadoop virtual cluster. However, when running hadoop MapReduce on a private cloud, the levels of control over the infrastructure are higher. As a consequence, in the future the hadoop is a expected workload to run in cloud virtual instances.

[3]https://github.com/brianfrankcooper/YCSB/wiki
[4]https://pypi.python.org/pypi/cloudbench/0.14.1

REFERENCES

ABD-EL-BARR, M.; EL-REWINI, H. **Fundamentals of Computer Organization and Architecture**. [S.l.]: Wiley, 2005. (Wiley Series on Parallel and Distributed Computing).

ABRAHAM, A. et al. **Advances in Computing and Communications, Part IV: first international conference, acc 2011, kochi, india, july 22-24, 2011. proceedings**. [S.l.]: Springer Berlin Heidelberg, 2011. n.pt. 4. (Communications in Computer and Information Science).

AGGARWAL, K. **Reliability Engineering**. [S.l.]: Springer Netherlands, 2012. (Topics in Safety, Reliability and Quality).

AIDA, K. et al. Evaluation on the Performance Fluctuation of Hadoop Jobs in the Cloud. In: COMPUTATIONAL SCIENCE AND ENGINEERING (CSE), 2013 IEEE 16TH INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2013. p.159–166.

AL-MUKHTAR, M. M. A.; MARDAN, A. A. A. Performance Evaluation of Private Clouds Eucalyptus versus CloudStack. **International Journal of Advanced Computer Science and Applications (IJACSA)**, [S.l.], v.5, n.5, 2014.

ALAMELDEEN, A. R.; WOOD, D. et al. Variability in architectural simulations of multi-threaded workloads. In: HIGH-PERFORMANCE COMPUTER ARCHITECTURE,

2003. HPCA-9 2003. PROCEEDINGS. THE NINTH INTERNATIONAL SYMPOSIUM ON. **Anais. . .** [S.l.: s.n.], 2003. p.7–18.

ALI-ELDIN, A.; TORDSSON, J.; ELMROTH, E. An adaptive hybrid elasticity controller for cloud infrastructures. In: NETWORK OPERATIONS AND MANAGEMENT SYMPOSIUM (NOMS), 2012 IEEE. **Anais. . .** [S.l.: s.n.], 2012. p.204–212.

ANTONOPOULOS, N.; GILLAM, L. **Cloud Computing: principles, systems and applications**. [S.l.]: Springer London, 2010. (Computer Communications and Networks).

APACHE. **Apache CloudStack Roadmap** <**https**://www.apache.org/>. Last access May, 2015.

ARMBRUST, M. et al. A view of cloud computing. **Communications of the ACM**, [S.l.], v.53, n.4, p.50–58, 2010.

ASHWINI, J.; DIVYA, C.; SANJAY, H. Efficient resource selection framework to enable cloud for HPC applications. In: COMPUTER AND COMMUNICATION TECHNOLOGY (ICCCT), 2013 4TH INTERNATIONAL CONFERENCE ON. **Anais. . .** [S.l.: s.n.], 2013. p.34–38.

AZODOLMOLKY, S.; WIEDER, P.; YAHYAPOUR, R. Cloud computing networking: challenges and opportunities for innovations. **Communications Magazine, IEEE**, [S.l.], v.51, n.7, p.54–62, 2013.

AZURE, M. **What is Azure,** <**http**://azure.microsoft.com/en-us/overview/what-is-azure//>. Last access Mar, 2015.

BAILEY, D. H. et al. The NAS parallel benchmarks. **International Journal of High Performance Computing Applications**, [S.l.], v.5, n.3, p.63–73, 1991.

BARKER, S. K.; SHENOY, P. Empirical evaluation of latency-sensitive application performance in the cloud. In: ACM SIGMM CONFERENCE ON MULTIMEDIA SYSTEMS. **Proceedings. . .** [S.l.: s.n.], 2010. p.35–46.

BAUN, C. et al. **Cloud Computing: web-based dynamic it services**. [S.l.]: Springer Berlin Heidelberg, 2011.

BIENIA, C. et al. The PARSEC benchmark suite: characterization and architectural implications. In: PARALLEL ARCHITECTURES AND COMPILATION TECHNIQUES, 17. **Proceedings...** [S.l.: s.n.], 2008. p.72–81.

BISSELING, R. **Parallel Scientific Computation:a structured approach using bsp and mpi: a structured approach using bsp and mpi**. [S.l.]: OUP Oxford, 2004. (Oxford scholarship online).

BOUTCHER, D.; CHANDRA, A. Does virtualization make disk scheduling passé? **ACM SIGOPS Operating Systems Review**, [S.l.], v.44, n.1, p.20–24, 2010.

BUCHTA, D.; EUL, M.; SCHULTE-CROONENBERG, H. **Strategic IT-Management: increase value, control performance, reduce costs**. [S.l.]: Gabler Verlag, 2010.

BUYYA, R.; BROBERG, J.; GOSCINSKI, A. **Cloud Computing: principles and paradigms**. [S.l.]: Wiley, 2010. (Wiley Series on Parallel and Distributed Computing).

BUYYA, R.; VECCHIOLA, C.; SELVI, S. **Mastering Cloud Computing**. [S.l.]: McGraw Hill, 2013.

Buyya, Rajkumar and Broberg, James and Goscinski, Andrzej M. **Cloud Computing Principles and Paradigms**. [S.l.]: Wiley Publishing, 2011.

CAMPESATO, O.; NILSON, K. **Web 2.0 Fundamentals: with ajax, development tools, and mobile platforms**. [S.l.]: Jones & Bartlett Learning, 2010.

CASTANO, V.; SCHAGAEV, I. **Resilient Computer System Design**. [S.l.]: Springer International Publishing, 2015.

CELESTI, A. et al. An Approach for the Composition of Generic Cloud-Based Services Using XRI-Based APIs for Enabling New E-Business. , [S.l.], 2013.

CERIER, A. et al. Infrastructure-as-a-service clouds in a professional environment. In: INFORMATION TECHNOLOGY EDUCATION, 2011. **Proceedings...** [S.l.: s.n.], 2011. p.301–302.

CEROTTI, D. et al. Flexible cpu provisioning in clouds: a new source of performance unpredictability. In: QUANTITATIVE EVALUATION OF SYSTEMS (QEST), 2012 NINTH INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2012. p.230–237.

CHANDRASEKARAN, K. **Essentials of Cloud Computing**. [S.l.]: Taylor & Francis, 2014.

CHEVANCE, R. **Server Architectures: multiprocessors, clusters, parallel systems, web servers, storage solutions**. [S.l.]: Elsevier Science, 2004.

CHOWDHURY, N. M. K.; BOUTABA, R. A survey of network virtualization. **Computer Networks**, [S.l.], v.54, n.5, p.862–876, 2010.

CLOUDSTACK. **Cloudstack Roadmap** <**https**://cloudstack.apache.org/>. Last access Mar, 2015.

CLOUDSTACK. **Apache CloudStack Features Roadmap** <**https**://cloudstack.apache.org/features.html>. Last access Jun, 2015.

COYNE, L. et al. **IBM Private, Public, and Hybrid Cloud Storage Solutions**. [S.l.]: IBM Redbooks, 2014.

CULLER, D.; SINGH, J.; GUPTA, A. **Parallel Computer Architecture: a hardware/software approach**. [S.l.]: Morgan Kaufmann Publishers, 1999. (The Morgan Kaufmann Series in Computer Architecture and Design Series).

DAI, Y. et al. Design and verification of a lightweight reliable virtual machine monitor for a many-core architecture. **Frontiers of Computer Science**, [S.l.], v.7, n.1, p.34–43, 2013.

DEAN, T. **Network+ Guide to Networks**. [S.l.]: Cengage Learning, 2012.

DEJUN, J.; PIERRE, G.; CHI, C.-H. EC2 performance analysis for resource provisioning of service-oriented applications. In: SERVICE-ORIENTED COMPUTING. ICSOC/SERVICEWAVE 2009 WORKSHOPS. **Anais...** [S.l.: s.n.], 2010. p.197–207.

DENTON, J. **Learning OpenStack Networking (Neutron)**. [S.l.]: Packt Publishing, 2014. (Community experience distilled).

DONGARRA, J.; LUSZCZEK, P.; PETITET, A. **The Linpack Benchmark: past, present and future**. [S.l.]: eeeee, 2001.

DUKARIC, R.; JURIC, M. B. Towards a unified taxonomy and architecture of cloud frameworks. **Future Generation Computer Systems**, [S.l.], v.29, n.5, p.1196–1210, 2013.

DUKARIĆ, R.; JURIČ, M. B. A Taxonomy and Survey of Infrastructure-as-a-Service Systems. **Lecture Notes on Information Theory Vol**, [S.l.], v.1, n.1, 2013.

EAST, I. **Computer Architecture And Organization**. [S.l.]: Taylor & Francis, 2004.

EGWUTUOHA, I. P. et al. A fault tolerance framework for high performance computing in cloud. In: CLUSTER, CLOUD AND GRID COMPUTING (CCGRID), 2012 12TH IEEE/ACM INTERNATIONAL SYMPOSIUM ON. **Anais...** [S.l.: s.n.], 2012. p.709–710.

EL-KHAMRA, Y. et al. Exploring the performance fluctuations of hpc workloads on clouds. In: CLOUD COMPUTING TECHNOLOGY AND SCIENCE (CLOUDCOM), 2010 IEEE SECOND INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2010. p.383–387.

ENDO, P. T. et al. A survey on open-source cloud computing solutions. In: **Anais...** [S.l.: s.n.], 2010. p.3–16.

EUCALYPTUS. **Eucalyptus: open source private cloud software,** <**https://www.eucalyptus.com/eucalyptus-cloud/iaas**>. Last access Mar, 2015.

EUCALYPTUS. **Walrus Storage**<**https://github.com/eucalyptus/eucalyptus/wiki/walrus** >. Last access jun, 2015.

EXPÓSITO, R. R. et al. Performance analysis of HPC applications in the cloud. **Future Generation Computer Systems**, [S.l.], v.29, n.1, p.218–229, 2013.

FATEMA, K. et al. A survey of Cloud monitoring tools: taxonomy, capabilities and objectives. **Journal of Parallel and Distributed Computing**, [S.l.], v.74, n.10, p.2918–2933, 2014.

FIELD, A.; MILES, J.; FIELD, Z. **Discovering Statistics Using R**. [S.l.]: SAGE Publications, 2012.

FIFIELD, T. et al. **OpenStack Operations Guide**. [S.l.]: O'Reilly Media, 2014.

FIORE, S.; ALOISIO, G. **Grid and Cloud Database Management**. [S.l.]: Springer Berlin Heidelberg, 2011. (SpringerLink : Bücher).

FORTIER, P.; MICHEL, H. **Computer Systems Performance Evaluation and Prediction**. [S.l.]: Digital Press, 2003. (Enterprise computing series).

FOWLER, F. **Survey Research Methods**. [S.l.]: SAGE Publications, 2008. (Applied Social Research Methods).

FURHT, B.; ESCALANTE, A. **Handbook of Cloud Computing**. [S.l.]: Springer US, 2010. (Computer science).

GHOSH, R. et al. Modeling and performance analysis of large scale IaaS Clouds. **Future Generation Computer Systems**, [S.l.], v.29, n.5, p.1216 – 1234, 2013. Special section: Hybrid Cloud Computing.

GHOSH, S. **Distributed Systems: an algorithmic approach, second edition**. [S.l.]: CRC Press, 2015. (Chapman & Hall/CRC Computer and Information Science Series).

GILLAM, L. et al. Fair benchmarking for cloud computing systems. **Journal of Cloud Computing: Advances, Systems and Applications**, [S.l.], v.2, n.1, p.6, 2013.

GOASGUEN, S. **60 Recipes for Apache CloudStack: using the cloudstack ecosystem**. [S.l.]: O'Reilly Media, 2014.

GOOGLE. **Google Compute Engine,** <**https**://cloud.google.com/compute/>. Last access Mar, 2015.

GOVINDAN, S. et al. Cuanta: quantifying effects of shared on-chip resource interference for consolidated virtual machines. In: ACM SYMPOSIUM ON CLOUD COMPUTING, 2. **Proceedings...** [S.l.: s.n.], 2011. p.22.

GROZEV, N.; BUYYA, R. Inter-Cloud architectures and application brokering: taxonomy and survey. **Software: Practice and Experience**, [S.l.], v.44, n.3, p.369–390, 2014.

GUENTER, B.; JAIN, N.; WILLIAMS, C. Managing cost, performance, and reliability tradeoffs for energy-aware server provisioning. In: INFOCOM, 2011 PROCEEDINGS IEEE. **Anais...** [S.l.: s.n.], 2011. p.1332–1340.

HAJJAT, M. et al. Application-specific configuration selection in the cloud: impact of provider policy and potential of systematic testing. , [S.l.], 2015.

HANSEN, P. **Operating System Principles**. [S.l.]: Prentice-Hall, 1973. (Prentice-Hall series in automatic computation).

HARING, G.; KOTSIS, G. **Computer Performance Evaluation: modelling techniques and tools. 7th international conference, vienna, austria, may 3 - 6, 1994. proceedings**. [S.l.]: Springer, 1994. (Lecture Notes in Computer Science).

HE, Q. et al. Case study for running HPC applications in public clouds. In: ACM INTERNATIONAL SYMPOSIUM ON HIGH PERFORMANCE DISTRIBUTED COMPUTING, 19. **Proceedings...** [S.l.: s.n.], 2010. p.395–401.

HENNESSY, J.; PATTERSON, D.; ASANOVIĆ, K. **Computer Architecture: a quantitative approach**. [S.l.]: Morgan Kaufmann/Elsevier, 2012. (Computer Architecture: A Quantitative Approach).

HENTGES, E. L.; THOMÉ, B. R. **Análise e Comparação de Ferramentas Open Source de Computação em Nuvem para o Modelo de Serviço IaaS**. 2013. Dissertação (Mestrado em Ciência da Computação) — Undergraduate Thesis, Sociedade Educacional Três de Maio (SETREM), Três de Maio, RS, Brazil.

HERKERSDORF, A.; RÖMER, K.; BRINKSCHULTE, U. **Architecture of Computing Systems - ARCS 2012: 25th international conference, munich, germany, february 28 - march 2, 2012. proceedings**. [S.l.]: Springer Berlin Heidelberg, 2012. (Lecture Notes in Computer Science / Theoretical Computer Science and General Issues).

HEROUX, M.; RAGHAVAN, P.; SIMON, H. **Parallel Processing for Scientific Computing**. [S.l.]: Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 2006. (SIAM e-books).

HOGGATT, J.; SHANK, J. **Century 21TM Computer Applications and Keyboarding, Lessons 1-170**. [S.l.]: Cengage Learning, 2009. (Business Presentation Series).

HP. **HP Cloud,** <**http**://www.hpcloud.com/>. Last access Mar, 2015.

HUANG, C.-T.; CARROLL, H.; PERRETTA, J. Improving Cloud Performance with Router-Based Filtering. In: **High Performance Cloud Auditing and Applications**. [S.l.]: Springer, 2014. p.323–343.

HUBER, N. et al. Evaluating and Modeling Virtualization Performance Overhead for Cloud Environments. In: CLOSER. **Anais...** [S.l.: s.n.], 2011. p.563–573.

HWANG, K.; DONGARRA, J.; FOX, G. **Distributed and Cloud Computing: from parallel processing to the internet of things**. [S.l.]: Elsevier Science, 2013.

IBM. **IBM IaaS Cloud,** <**http**://www.ibm.com/cloud-computing/us/en/iaas.html>. Last access Mar, 2015.

IOSUP, A.; PRODAN, R.; EPEMA, D. Iaas cloud benchmarking: approaches, challenges, and experience. In: **Cloud Computing for Data-Intensive Applications**. [S.l.]: Springer, 2014. p.83–104.

IOSUP, A.; YIGITBASI, N.; EPEMA, D. On the performance variability of production cloud services. In: CLUSTER, CLOUD AND GRID COMPUTING (CCGRID), 2011 11TH IEEE/ACM INTERNATIONAL SYMPOSIUM ON. **Anais...** [S.l.: s.n.], 2011. p.104–113.

IOSUP, A.; YIGITBASI, N.; EPEMA, D. On the performance variability of production cloud services. In: CLUSTER, CLOUD AND GRID COMPUTING (CCGRID), 2011 11TH IEEE/ACM INTERNATIONAL SYMPOSIUM ON. **Anais...** [S.l.: s.n.], 2011. p.104–113.

IOZONE. **Iozone File System Benchmark (Official Page)** <**http://iozone.org/**>. Last access in April, 2015.

IPERF. **What is Iperf?** <**http://iperf.fr/**>. Last access in April, 2015.

KANOUN, K.; SPAINHOWER, L. **Dependability Benchmarking for Computer Systems**. [S.l.]: Wiley, 2008. (Practitioners).

KAVIS, M. **Architecting the Cloud: design decisions for cloud computing service models (saas, paas, and iaas)**. [S.l.]: Wiley, 2014. (Wiley CIO).

KIVITY, A. et al. OSv—optimizing the operating system for virtual machines. In: 2014 USENIX ANNUAL TECHNICAL CONFERENCE (USENIX ATC 14). **Anais...** [S.l.: s.n.], 2014. v.1, p.61–72.

KOVARI, A.; DUKAN, P. KVM amp; OpenVZ virtualization based IaaS open source cloud virtualization platforms: opennode, proxmox ve. In: Intelligent Systems and Informatics (SISY), 2012 IEEE 10th Jubilee International Symposium on. **Anais...** [S.l.: s.n.], 2012. p.335–339.

KOZIEROK, C. **The TCP/IP Guide: a comprehensive, illustrated internet protocols reference**. [S.l.]: No Starch Press, 2005. (No Starch Press Series).

KUKUNAS, J. **Power and Performance: software analysis and optimization**. [S.l.]: Elsevier Science, 2015.

KUMAR, R. et al. Open Source Solution for Cloud Computing Platform Using OpenStack. **International Journal of Computer Science and Mobile Computing**, [S.l.], v.3, n.5, p.89–98, 2014.

KUMAR, R. et al. Apache CloudStack: open source infrastructure as a service cloud computing platform. In: INTERNATIONAL JOURNAL OF ADVANCEMENT IN ENGINEERING TECHNOLOGY, MANAGEMENT AND APPLIED SCIENCE. **Anais...** [S.l.: s.n.], 2014.

KUMAR, R.; GUPTA, S. Open Source Infrastructure for Cloud Computing Platform Using Eucalyptus. **Global Journal of Computers & Technology Vol**, [S.l.], v.1, n.2, 2014.

LEITNER, P.; CITO, J. Patterns in the Chaos-a Study of Performance Variation and Predictability in Public IaaS Clouds. **arXiv preprint arXiv:1411.2429**, [S.l.], 2014.

LIBVIRT. **Libvirt, the virtualization API,** <**https://libvirt.org/index.html**>. Last access jun, 2015.

LILJA, D. **Measuring Computer Performance: a practitioner's guide**. [S.l.]: Cambridge University Press, 2005.

LIU, F. et al. NIST cloud computing reference architecture. **NIST special publication**, [S.l.], v.500, p.292, 2011.

LIU, H. **Software Performance and Scalability: a quantitative approach**. [S.l.]: Wiley, 2011. (Quantitative Software Engineering Series).

LIU, X.; CHEN, J.; YANG, Y. **Temporal QOS Management in Scientific Cloud Workflow Systems**. [S.l.]: Elsevier Science, 2012. (Elsevier insights).

LOCAWEB. **Locaweb cloud<http://www.locaweb.com.br/cloud.html>**. Last access jun, 2015.

LUCAS-SIMARRO, J. L. et al. Scheduling strategies for optimal service deployment across multiple clouds. **Future Generation Computer Systems**, [S.l.], v.29, n.6, p.1431–1441, 2013.

MAHMOOD, Z. **Continued Rise of the Cloud: advances and trends in cloud computing**. [S.l.]: Springer London, 2014. (Computer Communications and Networks).

MARATHE, A. et al. A comparative study of high-performance computing on the cloud. In: HIGH-PERFORMANCE PARALLEL AND DISTRIBUTED COMPUTING, 22. **Proceedings...** [S.l.: s.n.], 2013. p.239–250.

MARINESCU, D. **Cloud Computing: theory and practice**. [S.l.]: Elsevier Science, 2013.

MARON, C. A. F. **Avaliação e Comparação da Computação de Alto Desempenho em Ferramentas Opensource de Administração de Nuvem Usando Estações De Trabalho**. 2014. Dissertação (Mestrado em Ciência da Computação) — Undergraduate Thesis, Sociedade Educacional Três de Maio (SETREM), Três de Maio, RS, Brazil.

MARON, C. A. F. et al. Avaliação e Comparação do Desempenho das Ferramentas OpenStack e OpenNebula. In: ESCOLA REGIONAL DE REDES DE COMPUTADORES (ERRC), 12., Canoas. **Anais...** Sociedade Brasileira de Computação, 2014. p.1–5.

MARON, C. A. F. et al. Em Direção à Comparação do Desempenho das Aplicações Paralelas nas Ferramentas OpenStack e OpenNebula. In: ESCOLA REGIONAL DE ALTO DESEMPENHO DO ESTADO DO RIO GRANDE DO SUL (ERAD/RS), 15., Gramado, RS, Brazil. **Anais...** Sociedade Brasileira de Computação, 2015.

MARON, C. A. F.; GRIEBLER, D.; SCHEPKE, C. Comparação das Ferramentas OpenNebula e OpenStack em Nuvem Composta de Estações de Trabalho. In: ESCOLA REGIONAL DE ALTO DESEMPENHO DO ESTADO DO RIO GRANDE DO SUL (ERAD/RS), 14., Alegrete, RS, Brazil. **Anais...** Sociedade Brasileira de Computação, 2014. p.173–176.

MATHEWS, J. **EXECUTANDO O XEN: um guia pratico para a arte da virtualizaçao**. [S.l.]: STARLIN ALTA CONSULT, 2009.

MATTSON, T.; SANDERS, B.; MASSINGILL, B. **Patterns for Parallel Programming**. [S.l.]: Pearson Education, 2004. (Software Patterns Series).

MAUCH, V.; KUNZE, M.; HILLENBRAND, M. High performance cloud computing. **Future Generation Computer Systems**, [S.l.], v.29, n.6, p.1408–1416, 2013.

MCCALPIN, J. **The Stream Benchmark)** <**https://www.cs.virginia.edu/stream/**>. Last access in April, 2015.

MCCOOL, M.; ROBISON, A.; REINDERS, J. **Structured Parallel Programming: patterns for efficient computation**. [S.l.]: Elsevier/Morgan Kaufmann, 2012. (Morgan Kaufmann).

MISHRA, A. K. et al. Towards characterizing cloud backend workloads: insights from google compute clusters. **ACM SIGMETRICS Performance Evaluation Review**, [S.l.], v.37, n.4, p.34–41, 2010.

NAMBIAR, R.; POESS, M. **Selected Topics in Performance Evaluation and Benchmarking: 4th tpc technology conference, tpctc 2012, istanbul, turkey, august 27, 2012, revised selected papers**. [S.l.]: Springer Berlin Heidelberg, 2013. (Lecture Notes in Computer Science).

NATHUJI, R.; KANSAL, A.; GHAFFARKHAH, A. Q-clouds: managing performance interference effects for qos-aware clouds. In: EUROPEAN CONFERENCE ON COMPUTER SYSTEMS, 5. **Proceedings...** [S.l.: s.n.], 2010. p.237–250.

NICOLAE, B.; RITEAU, P.; KEAHEY, K. Transparent Throughput Elasticity for IaaS Cloud Storage Using Guest-Side Block-Level Caching. In: UTILITY AND CLOUD COMPUTING (UCC), 2014 IEEE/ACM 7TH INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2014. p.186–195.

NIMBUS. **About Nimbus,** <**http://www.nimbusproject.org/about**>. Last access Mar, 2015.

NOVAKOVIC, D. et al. **Deepdive: transparently identifying and managing performance interference in virtualized environments**. [S.l.: s.n.], 2013.

OLUKOTUN, K. et al. **Chip Multiprocessor Architecture: techniques to improve throughput and latency**. [S.l.]: Morgan & Claypool Publishers, 2007. (Synthesis lectures in computer architecture).

OPENNEBULA. **OpenNebula Roadmap** <**http**://opennebula.org//>. Last access Mar, 2015.

OPENNEBULA. **OpenNebula Scheduler** <**http://docs.opennebula.org/4.12/administration/refer**
Last access jun, 2015.

OPENQRM. **OpenQRM Community Edition,** <**http://www.openqrm-enterprise.com/products/community-edition.html**>. Last access Mar, 2015.

OPENQRM. **OpenQRM Overview,** <**http://sourceforge.net/projects/openqrm/add-ons-plugins**>. Last access Jun, 2015.

OPENSTACK. **OpenStack Roadmap** <**http**://openstack.org/software/roadmap/>. Last access Mar, 2015.

OPENSTACK. **OpenStack Docs Roadmap** <**http**://docs.openstack.org/admin-guide-cloud/content/>. Last access May, 2015.

OPENSTACK. **OpenStack API Reference Documentation** <**http://docs.openstack.org/api/api-ref-guides.html**>. Last access jun, 2015.

OSV. **OSv**<**http://osv.io/**>. Last access jun, 2015.

PARSONS, J.; OJA, D. **New Perspectives on Computer Concepts, Comprehensive**. [S.l.]: Cengage Learning, 2008. (Available Titles Skills Assessment Manager (SAM) - Office 2007 Series).

PATTERSON, D.; HENNESSY, J. **Computer Organization and Design: the hardware/software interface**. [S.l.]: Elsevier Science, 2013. (The Morgan Kaufmann Series in Computer Architecture and Design).

PENG, J. et al. Comparison of several cloud computing platforms. In: INFORMATION SCIENCE AND ENGINEERING (ISISE), 2009 SECOND INTERNATIONAL SYMPOSIUM ON. **Anais...** [S.l.: s.n.], 2009. p.23–27.

PRODAN, R.; OSTERMANN, S. A survey and taxonomy of infrastructure as a service and web hosting cloud providers. In: GRID COMPUTING, 2009 10TH IEEE/ACM INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2009. p.17–25.

PU, X. et al. Understanding performance interference of i/o workload in virtualized cloud environments. In: CLOUD COMPUTING (CLOUD), 2010 IEEE 3RD INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2010. p.51–58.

RABBITMQ. **RabbitMQ**<**https://www.rabbitmq.com/**>. Last access jun, 2015.

RADHAKRISHNAN, S. et al. NicPic: scalable and accurate end-host rate limiting. In: PRESENTED AS PART OF THE 5TH USENIX WORKSHOP ON HOT TOPICS IN CLOUD COMPUTING. **Anais...** [S.l.: s.n.], 2013.

RAJ, H. et al. Self virtualized IO: high performance, scalable io virtualization in multicore systems. , [S.l.], 2006.

RAJ, H. et al. Resource Management for Isolation Enhanced Cloud Services. In: ACM WORKSHOP ON CLOUD COMPUTING SECURITY, 2009., New York, NY, USA. **Proceedings...** ACM, 2009. p.77–84. (CCSW '09).

REDDY, P. V. V.; RAJAMANI, L. Performance Comparison of Hypervisors in the Private Cloud. In: **Advanced Computing, Networking and Informatics-Volume 2**. [S.l.]: Springer, 2014. p.393–402.

REED, D.; FUJIMOTO, R. **Multicomputer Networks: message-based parallel processing**. [S.l.]: MIT Press, 2004. (Scientific Computation).

REESE, G. **Cloud Application Architectures: building applications and infrastructure in the cloud**. [S.l.]: O'Reilly Media, 2009. (Theory in Practice (O'Reilly)).

REGOLA, N.; DUCON, J. C. Recommendations for Virtualization Technologies in High Performance Computing. In: ACM, Indianapolis. **Anais...** IEEE, 2010. p.409–416.

RIEGELMAN, R. **Studying a Study and Testing a Test: how to read the medical evidence**. [S.l.]: Lippincott Williams & Wilkins, 2005. (Core Handbook Series in Pediatrics Series).

ROEBUCK, K. **NFS - Network File System: high-impact strategies - what you need to know: definitions, adoptions, impact, benefits, maturity, vendors**. [S.l.]: Emereo Publishing, 2012.

ROLOFF, E. et al. High Performance Computing in the cloud: deployment, performance and cost efficiency. **CloudCom**, [S.l.], v.2012, p.371–378, 2012.

ROVEDA, D.; VOGEL, A.; GRIEBLER, D. Understanding, Discussing and Analyzing the OpenNebula and OpenStack's IaaS Management Layers. **Revista Eletrônica Argentina-Brasil de Tecnologias da Informação e da Comunicação**, Três de Maio, Brazil, v.3, n.1, p.15, August 2015.

RUSSELL, J.; COHN, R. **Phoronix Test Suite**. [S.l.]: Book on Demand, 2012.

SABHARWAL, N. **Apache CloudStack Cloud Computing**. [S.l.]: Packt Publishing, 2013. (Community experience distilled).

SAKR, S.; GABER, M. **Large Scale and Big Data: processing and management**. [S.l.]: Taylor & Francis, 2014. (An Auerbach book).

SANTOS, J. R. et al. Bridging the Gap between Software and Hardware Techniques for I/O Virtualization. In: USENIX ANNUAL TECHNICAL CONFERENCE. **Anais...** [S.l.: s.n.], 2008. p.29–42.

SCHAD, J.; DITTRICH, J.; QUIANÉ-RUIZ, J.-A. Runtime measurements in the cloud: observing, analyzing, and reducing variance. **Proceedings of the VLDB Endowment**, [S.l.], v.3, n.1-2, p.460–471, 2010.

SCHEUNER, J. et al. Cloud Work Bench–Infrastructure-as-Code Based Cloud Benchmarking. In: CLOUD COMPUTING TECHNOLOGY AND SCIENCE (CLOUDCOM), 2014 IEEE 6TH INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2014. p.246–253.

SERRANO, N.; GALLARDO, G.; HERNANTES, J. Infrastructure as a Service and Cloud Technologies. **IEEE Software**, [S.l.], n.2, p.30–36, 2015.

SHIN, H.; WIN, M. Z. Gallager's exponent for MIMO channels: a reliability-rate tradeoff. **Communications, IEEE Transactions on**, [S.l.], v.57, n.4, p.972–985, 2009.

SHROFF, G. **Enterprise Cloud Computing: technology, architecture, applications**. [S.l.]: Cambridge University Press, 2010.

SILVA, M. et al. CloudBench: experiment automation for cloud environments. In: CLOUD ENGINEERING (IC2E), 2013 IEEE INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2013. p.302–311.

SOSINSKY, B. **Cloud Computing Bible**. [S.l.]: Wiley, 2010.

SOTOMAYOR, B. et al. Virtual infrastructure management in private and hybrid clouds. **Internet computing, IEEE**, [S.l.], v.13, n.5, p.14–22, 2009.

SRIVASTAVA, S.; CIORBA, F. M.; BANICESCU, I. Employing a study of the robustness metrics to assess the reliability of dynamic loop scheduling. , [S.l.], 2011.

STERLING, T. **Beowulf Cluster Computing with Linux**. [S.l.]: MIT Press, 2002. (Scientific and Computational Engineering Series).

SUN, A. et al. IaaS public cloud computing platform scheduling model and optimization analysis. **Int'l J. of Communications, Network and System Sciences**, [S.l.], v.4, n.12, p.803, 2011.

SUN, B. et al. Benchmarking Private Cloud Performance with User Centric Metrics. In: CLOUD ENGINEERING (IC2E), 2014 IEEE INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2014. p.311–318.

TAFA, I. et al. The evaluation of transfer time, cpu consumption and memory utilization in xen-pv, xen-hvm, openvz, kvm-fv and kvm-pv hypervisors using ftp and http approaches. In: INTELLIGENT NETWORKING AND COLLABORATIVE SYSTEMS (INCOS), 2011 THIRD INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2011. p.502–507.

TANENBAUM, A. **Distributed Operating Systems**. [S.l.]: Pearson Education, 1995.

TANG, C. FVD: a high-performance virtual machine image format for cloud. In: USENIX ANNUAL TECHNICAL CONFERENCE. **Anais...** [S.l.: s.n.], 2011.

THOME, B.; HENTGES, E.; GRIEBLER, D. Computação em Nuvem: análise comparativa de ferramentas open source para iaas. In: ESCOLA REGIONAL DE REDES DE COMPUTADORES (ERRC), 11., Porto Alegre, RS, Brazil. **Anais...** Sociedade Brasileira de Computação, 2013. p.4.

TIAN, J.; RUDRARAJU, S.; LI, Z. Evaluating web software reliability based on workload and failure data extracted from server logs. **Software Engineering, IEEE Transactions on**, [S.l.], v.30, n.11, p.754–769, 2004.

TOOSI, A. N.; CALHEIROS, R. N.; BUYYA, R. Interconnected cloud computing environments: challenges, taxonomy, and survey. **ACM Computing Surveys (CSUR)**, [S.l.], v.47, n.1, p.7, 2014.

TORALDO, G. **OpenNebula 3 Cloud Computing**. [S.l.]: Packt Publishing, 2012. (Community experience distilled).

TSUGAWA, M.; MATSUNAGA, A.; FORTES, J. A. Cloud Networking to Support Data Intensive Applications. In: **Cloud Computing for Data-Intensive Applications**. [S.l.]: Springer, 2014. p.61–81.

TUDORAN, R. et al. A performance evaluation of Azure and Nimbus clouds for scientific applications. In: INTERNATIONAL WORKSHOP ON CLOUD COMPUTING PLATFORMS, 2. **Proceedings...** [S.l.: s.n.], 2012. p.4.

VAQUERO, L. **Open Source Cloud Computing Systems: practices and paradigms: practices and paradigms**. [S.l.]: Information Science Reference, 2012. (Premier reference source).

VICAT-BLANC, P. et al. **Computing Networks: from cluster to cloud computing**. [S.l.]: Wiley, 2013. (ISTE).

VIDYARTHI, D. et al. **Scheduling in Distributed Computing Systems**: analysis, design and models. [S.l.]: Springer US, 2008.

WANG, G.; NG, T. E. The impact of virtualization on network performance of amazon ec2 data center. In: INFOCOM, 2010 PROCEEDINGS IEEE. **Anais...** [S.l.: s.n.], 2010. p.1–9.

WARD, J. S.; BARKER, A. Observing the clouds: a survey and taxonomy of cloud monitoring. **Journal of Cloud Computing**, [S.l.], v.3, n.1, p.1–30, 2014.

WEBER, S. **The Success of Open Source**. [S.l.]: Harvard University Press, 2009.

WHITE, C. **Data Communications and Computer Networks: a business user's approach**. [S.l.]: Cengage Learning, 2012.

WILLIAMS, M. **A Quick Start Guide to Cloud Computing: moving your business into the cloud**. [S.l.]: Kogan Page, 2010. (New Tools for Business).

WILLMANN, P. et al. Concurrent direct network access for virtual machine monitors. In: HIGH PERFORMANCE COMPUTER ARCHITECTURE, 2007. HPCA 2007. IEEE 13TH INTERNATIONAL SYMPOSIUM ON. **Anais...** [S.l.: s.n.], 2007. p.306–317.

WIND, S. Open source cloud computing management platforms: introduction, comparison, and recommendations for implementation. In: OPEN SYSTEMS (ICOS), 2011 IEEE CONFERENCE ON. **Anais...** [S.l.: s.n.], 2011. p.175–179.

WOLSKI, R.; BREVIK, J. Using parametric models to represent private cloud workloads. **Services Computing, IEEE Transactions on**, [S.l.], v.7, n.4, p.714–725, 2014.

WU, T.-L.; BAASHA, S. N.; KARWA, S. S. A Survey of Open Source Cloud Infrastructure using Future Grid Testbed. , Bloomington, 2010.

XAVIER, M. et al. Performance Evalution of Container-based Virtualization for High Performance Computing Environments. In: EUROMICRO INTERNATIONAL CONFERENCE ON PARALLEL, DISTRIBUTED AND NETWORK-BASED PROCESSING (PDP), 21., Belfast, UK. **Anais...** IEEE, 2013. p.233–240.

XU, F. et al. Managing performance overhead of virtual machines in cloud computing: a survey, state of the art, and future directions. **Proceedings of the IEEE**, [S.l.], v.102, n.1, p.11–31, 2014.

XU, G.; PANG, J.; FU, X. A load balancing model based on cloud partitioning for the public cloud. **Tsinghua Science and Technology**, [S.l.], v.18, n.1, p.34–39, 2013.

YADAV, S. Comparative Study on Open Source Software for Cloud Computing Platform: eucalyptus, openstack and opennebula. **International Journal Of Engineering And Science**, [S.l.], v.3, n.10, p.51–54, 2013.

YE, K. et al. Live migration of multiple virtual machines with resource reservation in cloud computing environments. In: CLOUD COMPUTING (CLOUD), 2011 IEEE INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2011. p.267–274.

YOUNGE, A. J. et al. Analysis of virtualization technologies for high performance computing environments. In: CLOUD COMPUTING (CLOUD), 2011 IEEE INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2011. p.9–16.

YSCB. **Yahoo! Cloud Serving Benchmark (YCSB)**<**https://github.com/brianfrankcooper/ycsb/w**
Last access jun, 2015.

ZHAO, L. et al. **Cloud Data Management**. [S.l.]: Springer International Publishing, 2014.

ZHU, Q.; TUNG, T. A performance interference model for managing consolidated workloads in QoS-aware clouds. In: CLOUD COMPUTING (CLOUD), 2012 IEEE 5TH INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2012. p.170–179.

ZILIAK, S.; MCCLOSKEY, D. **The Cult of Statistical Significance: how the standard error costs us jobs, justice, and lives**. [S.l.]: University of Michigan Press, 2008. (Economics, cognition, and society).