UNIVERSITY OF ZAGREB
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

MASTER THESIS n. 721

PREDICTIVE MODEL OF FAILURES IN CLOUD COMPUTING

Marija Mijić

Zagreb, June 2014.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA
ODBOR ZA DIPLOMSKI RAD PROFILA

Zagreb, 10. ožujka 2014.

DIPLOMSKI ZADATAK br. 721

Pristupnik: **Marija Mijić (0036449033)**
Studij: Informacijska i komunikacijska tehnologija
Profil: Telekomunikacije i informatika

Zadatak: **Model predviđanja prekida rada kod računarstva u oblaku**

Opis zadatka:

Usluga računarstva u oblaku je omogućila tvrtkama da povjere održavanje svoje informacijske infrastrukture te upravljanje pohranom podataka specijaliziranim pružateljima takvih usluga. U poslovnom svijetu su široko prepoznate prednosti tog koncepta te broj pružatelja i korisnika računarstva u oblaku raste iz dana u dan. Međutim, predviđanja pokazuju da će svoj vrhunac računarstvo u oblaku tek doživjeti za nekoliko godina. Stoga je pouzdanost usluge računarstva u oblaku sve važnije područje istraživanja, budući da korisnici usluge najčešće niti ne znaju gdje je infrastruktura koja pruža uslugu fizički smještena.
Vaša zadaća je unaprijediti način procjene otpornosti usluge računarstva u oblaku. Posebnu pozornost stavite na analizu prekida rada i nedostupnosti poslužitelja na kojima je usluga fizički smještena. Vaš prvi zadatak je istražiti učestalost i uzroke takvih prekida te izraditi model za predviđanje istih. Vaš drugi zadatak je odrediti metodologiju i metrike za procjenu performansi različitih pružatelja usluge računarstva u oblaku te na temelju tih metrika usporediti neke od pružatelja usluge.
Svu potrebnu literaturu i uvjete za rad osigurat će Vam Zavod za telekomunikacije.

Zadatak uručen pristupniku: 14. ožujka 2014.
Rok za predaju rada: 30. lipnja 2014.

Mentor:

Doc.dr.sc. Vedran Podobnik

Djelovođa:

Doc.dr.sc. Lea Skorin-Kapov

Predsjednik odbora za diplomski rad profila:

Prof.dr.sc. Igor Sunday Pandžić

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA
ODBOR ZA DIPLOMSKI RAD PROFILA

Zagreb, 10. ožujka 2014.

DIPLOMSKI ZADATAK br. 721

Pristupnik: **Marija Mijić (0036449033)**
Studij: Informacijska i komunikacijska tehnologija
Profil: Telekomunikacije i informatika

Zadatak: **Predictive Model of Failures in Cloud Computing**

Opis zadatka:

Cloud computing services have enabled companies to outsource the maintenance of its information infrastructure and data storage management to specialized providers of such services. Advantages of this approach are widely recognized in the business world. While the number of both providers and users of cloud computing is growing on a daily basis, forecasts show that cloud computing will experience its peak in years to come. Therefore, reliability of cloud computing services is increasingly important area of research, as service users often do not know where the infrastructure that provides services is physically located.

Your assignment is to improve the way how the resilience of cloud computing services is estimated. Pay a special attention to the analysis of the failures and unavailability of servers where cloud computing services are physically located. Your first task is to explore the frequency and causes of such failures and to build a predictive model of failures in cloud computing. Your second task is firstly to define the methodology and metrics to evaluate the performance of various cloud computing service providers and afterwards to compare a set of cloud computing service providers based on the established metrics.

All of the resources will be provided to you by the Department of Telecommunications.

Zadatak uručen pristupniku: 14. ožujka 2014.
Rok za predaju rada: 30. lipnja 2014.

Mentor:

Doc.dr.sc. Vedran Podobnik

Djelovođa:

Doc.dr.sc. Lea Skorin-Kapov

Predsjednik odbora za diplomski rad profila:

Prof.dr.sc. Igor Sunday Pandžić

To Mom, Dad and Ana

Contents

Introduction

Cloud computing industry has been growing since its creation. For many businesses this meant being able to fully focus on their core activities. It also meant that they would no longer have to buy expensive machines before they even needed them. Cloud computing is as much used by small companies as it is used by multinational corporations. And the transformation from a small business to a bigger one is supported by the cloud computing though its scalability. However, the current state of cloud computing, given the numerous outages, data loss and security breaches, does not justify its status of the future of information technology. One of the main problems of cloud computing lies in the fact that the cloud is centralized. Modern data centers are as big as factories and often difficult to supervise. This leaves cloud computing prone to attacks and unfit for fast recovery. In other words, the cloud is not resilient. Still, the statistics of the cloud failures are not widely known across the industry and the evaluation of the cloud still fails to generate a ranking list of cloud computing providers that would benefit both customers (when choosing the providers) and the providers (to know in which direction they should try to improve). In the Thesis, a new model of cloud computing will be used to improve its availability tracking and resilience.

The first chapter explains some basic terms and concepts of the cloud computing. The second chapter contains the research done on the evaluation of current state of the cloud. It also provides an insight into the downtime statistics of the cloud computing providers as well as their ranking list. The third chapter explains how the evaluation of the cloud computing providers has been done so far. The fifth chapter proposes a new model of evaluation and ranking of the cloud computing providers. The last chapter shows the results of the testing of virtual machines of some of the biggest cloud computing providers today and explains how the results obtained from the tests can be used to rank and evaluate providers.

1. Cloud Computing

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models [1].

Many argue that the cloud had existed long before the actual term 'cloud' was coined. The cloud refers to both datacenter hardware and software and although the development of extremely large scale datacenters is considered to be the main reason of such wide use of the cloud today, there are other factors that played a big part in the global acceptance of cloud computing. Such factors, such as additional technology trends and new business models, made new application opportunities and usage models possible. Therefore, even though similar concepts existed, the usage was not quite the same as it is that of what is today considered to be cloud. Moreover, new application opportunities render cloud use more convenient than it has ever been.

New Technology Trends and Business Models:

1. <u>Credit cards accepted online</u>

   Most of the cloud computing providers today offer scalable resources, "pay-as-you-go" model, which means that the cloud is used when needed and paid for when used. Such services could not be provided were it necessary to create a contract every time a client wanted to use more resources. In addition to that, small businesses that previously could not afford long-term commitments with payment processing service such as VeriSign or Authorize.net can now use low-cost card payment processing services such as PayPal.

2. <u>Hardware-level virtualization</u>

   The second innovation that helped the creation of cloud computing is the hardware-level virtualization that allows customers to choose their own stack without disrupting each other. That meant that the customers could now share the same hardware and, therefore further lower the costs.

New Application Opportunities

1. Mobile interactive applications

   Mobile phone sales grew 46.5 percent in second quarter of 2013 and exceeded feature phone sales for first time [2]. Mobile applications are now making up for as much as 86% of overall time spent on mobile device [3]. With mobile phone battery use being one of the biggest challenges and wide-area networking relatively higher than any other IT costs [4], mobile interactive applications, that require high availability and at the same time generate a large amount of data, will be able to use cloud to store data at a convenient distance. In addition to that, by removing some of the computation that is nowadays done by the phone itself and delegating them to the cloud, battery may last significantly longer. However, the traditional cloud model needs to be modified in order to adapt to these new services. Cloudlets are one of the suggested models for this type of use of the cloud. A cloudlet is a new architectural element that arises from the convergence of mobile computing and cloud computing. It represents the middle tier of a 3-tier hierarchy: mobile device --- cloudlet --- cloud. A cloudlet can be viewed as a "data center in a box" whose goal is to "bring the cloud closer". A Cloudlet is viewed in the very recent literature as a WiFi access-point used as an offloading facility for the mobile users. It opens the way to almost instantaneous mobile offloading. This property is of the highest importance for the mobile terminals manufacturers.

2. Parallel batch processing

   Cloud computing made fast data processing available even to small businesses. Contrary to the model where a company had to purchase costly machines that, in many cases, were going to be used occasionally, nowadays it is possible to leverage existing machines of other companies (such as Google's App Engine or Amazon Elastic Compute Cloud) and pay only for the time machines were used.

3. The rise of analytics

   With parallel batch processing becoming so wide-spread, there is a rise in the analysis of the data and decision support systems that require the use of extensive computer resources. A use of such systems is found in trying to

understand customer behaviors supply chains, buying habits, ranking, and so on.

4. Extension of compute-intensive desktop applications

   Some of the most popular computation tools (such as Matlab and Mathematica) have integrated the cloud into their environment. It serves as an extension used to perform expensive evaluations. Other applications could possibly benefit from the cloud following the same path. The latest versions of the mathematics software packages Matlab and Mathematica are capable of using Cloud Computing to perform expensive evaluations [5].

5. "Earthbound" applications

   There are, however, still some obstacles that render some applications inapt for user of the cloud. The cost of wide-area data transfer is still relatively hard, so, for applications that deal with a lot of data, it could be more expensive to move the data then to use their own equipment to do the computation. In addition, some applications require microsecond precision (for example, stock trading) and, considering the latency of wide-area data transfer, cloud cannot be used in such cases.

1.1. Essential Characteristics of Cloud Computing

On-demand self-service.

The contracts between a customer and a provider are automated with no need for human interaction. Similarly, the service can be stopped at any time, in case the customer decided they will not need the cloud anymore.

Broad network access.

The cloud is accessible over the network and can be accessed through think or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations).

Resource pooling.

The providers' resources are shared by multiple users using a concept called multitenancy. Multitenancy is a reference to the mode of operation of software where multiple independent instances of one or multiple applications operate in a shared environment. The

instances (tenants) are logically isolated, but physically integrated. The degree of logical isolation must be complete, but the degree of physical integration will vary [6]. The users are unaware of the physical location of the resources as well as of the existence of other tenant sharing the same location.

Rapid elasticity.

Allocated resources can be elastically provisioned and released. From the customers' perspective, it means there is no need to allocate more than they need at the moment of signing the contract. If their needs later on turn out to be growing, the customers can easily increase the resources as they appear unlimited to them.

Measured service.

The usage of capabilities is measured and can be thus automatically controlled and optimized.

1.2. Service Models

There are three service models in cloud computing: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). The difference between those models lies in the amount of responsibilities that are shifted to the cloud provider, depending of the needs of the customers.

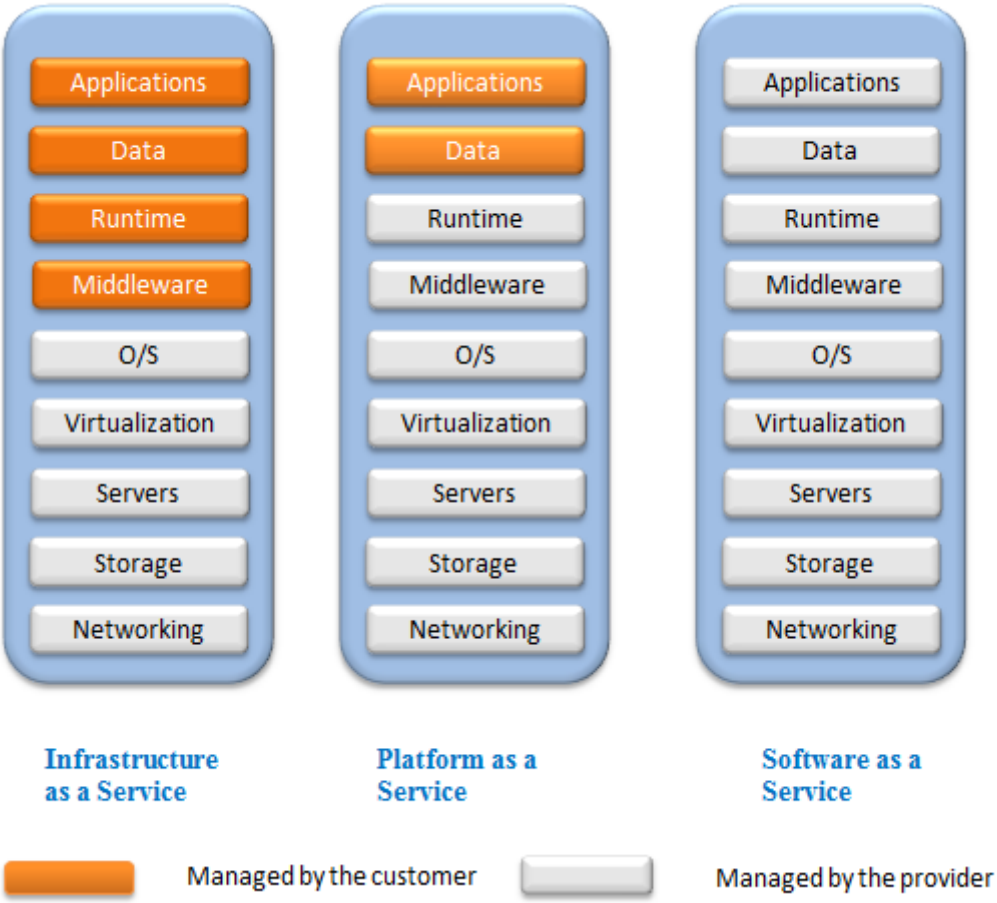


Figure 1.1 A Comparison of Cloud Computing Models

Figure 1.1 shows an overview of different cloud computing models. The orange cells represent the part the is managed by the customer and the grey cells are managed by the providers.

Software as a Service (SaaS).

SaaS can be defined as "software deployed as a hosted service and accessed over the Internet" [7]. The provider manages the cloud's entire infrastructure, including network, servers, operating systems, storage and the applications. An example of SaaS would be SalesForce.com. SalesForce.com is best known for its Customer Relationship Management

(CRM) application. The main purpose of the application is to track all sales activity. Customers can start using the application once they have subscribed, while SalesForce.com takes care of the infrastructure and the application itself. Other examples of SaaS providers include NetSuite, Oracle, IBM, and Microsoft. This service model is most adapted for companies that want to focus more on their own business requirements and less on the underlying technologies.

Platform as a Service (PaaS).

By choosing this service model, customers get a higher level of autonomy than they would get by using SaaS. The provider is still, however, in charge of the cloud infrastructure including network, servers, operating systems and storage, but customers can deploy their own applications and configure their application-hosting environment. Creating and maintaining an infrastructure is the most time-consuming work in the on-premises systems. PaaS was invented to solve exactly this problem [8]. Some of the most popular PaaS providers are Google AppEngine, Windows Azure, Heroku, Red Hat OpenShift, etc.

Infrastructure as a Service (IaaS).

While PaaS allows customers to deploy their own applications, for some users that might not be enough. The PaaS vs. IaaS decision typically is determined by the performance and scalability requirements of the application. PaaS solutions have limitations on their ability to achieve very high scale due to the fact that these platforms must provide auto scaling and failover capabilities for all tenants of the platform [9]. In IaaS model, the provider provisions processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. Key examples are Amazon AWS, Windows Azure, Google Compute Engine, Rackspace Open Cloud, IBM SmartCloud Enterprise, HP Enterprise Converged Infrastructure, etc.

1.3. Deployment Models

Private cloud.

The infrastructure of a private cloud is provisioned for the use by a single organization, including multiple consumers. This model is less restrictive since the data and processes are managed within the organization, so there are no restrictions of network bandwidth, security exposures or legal requirements that using public cloud across open, public networks might entail. The infrastructure of this type of cloud may be owned, managed and operated by the organization that uses it, a third party or a combination of them. In addition, it may be deployed on and off organization premises. Examples of private cloud include Amazon VPC, Eucalyptus, Enomaly, VMWare, Redplaid, Platform computing, etc.

Community cloud.

Community cloud is in a way an extension of private cloud. The cloud is used by a specific community of consumers from organizations that have shared concerns. One or more organizations in the community may be in charge or owning, managing and operating the cloud, but that role can also belong to a third party or some combination of community members and the third party. Just like the private cloud, community cloud may exist on and off community premises.

Public cloud.

The cloud infrastructure is open for use by the general public. Public cloud describes cloud computing in the traditional mainstream sense, where the resources are provisioned over the Internet, via applications and web services. It is owned and managed by a business, academic or government organization or some combination of them. Examples of public loud provides include Amazon, Windows Azure, Rackspace, IBM, Joyent, etc.

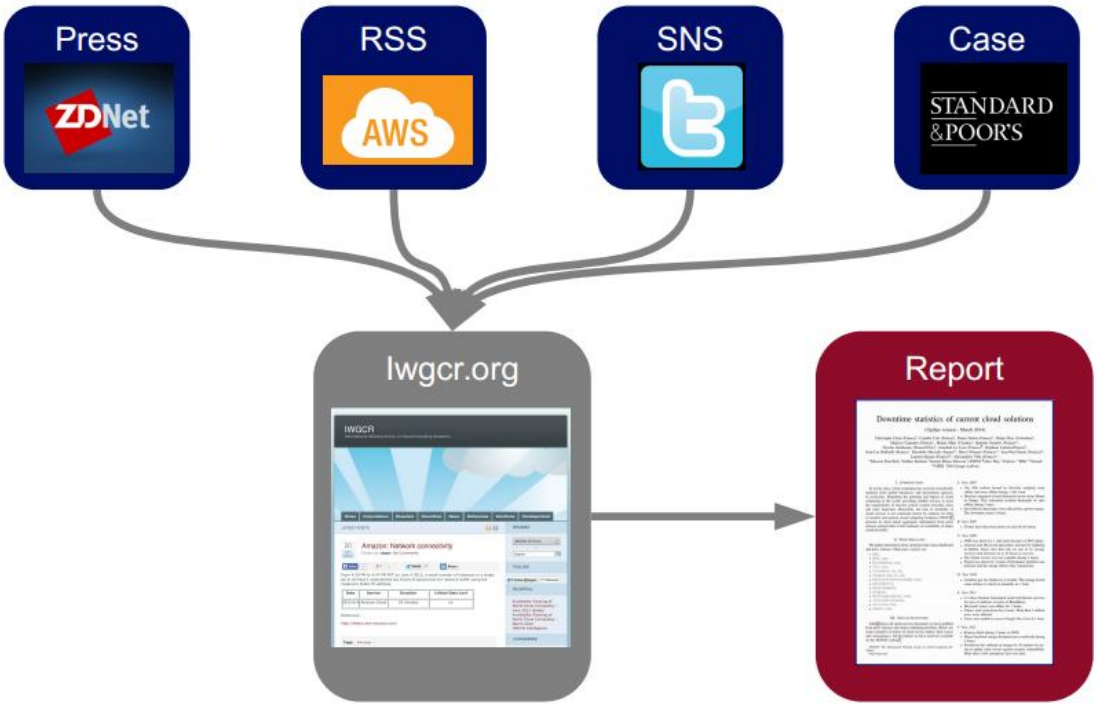
Hybrid cloud.

There are multiple internal and/or external providers in the environment of hybrid cloud. The providers are bound together by standardized or proprietary technology that enables

data and application portability. Examples include RightScale, Asigra Hybrid Cloud Backup, Carpathia, Skytap, etc.

1.4. Current State of Cloud Computing

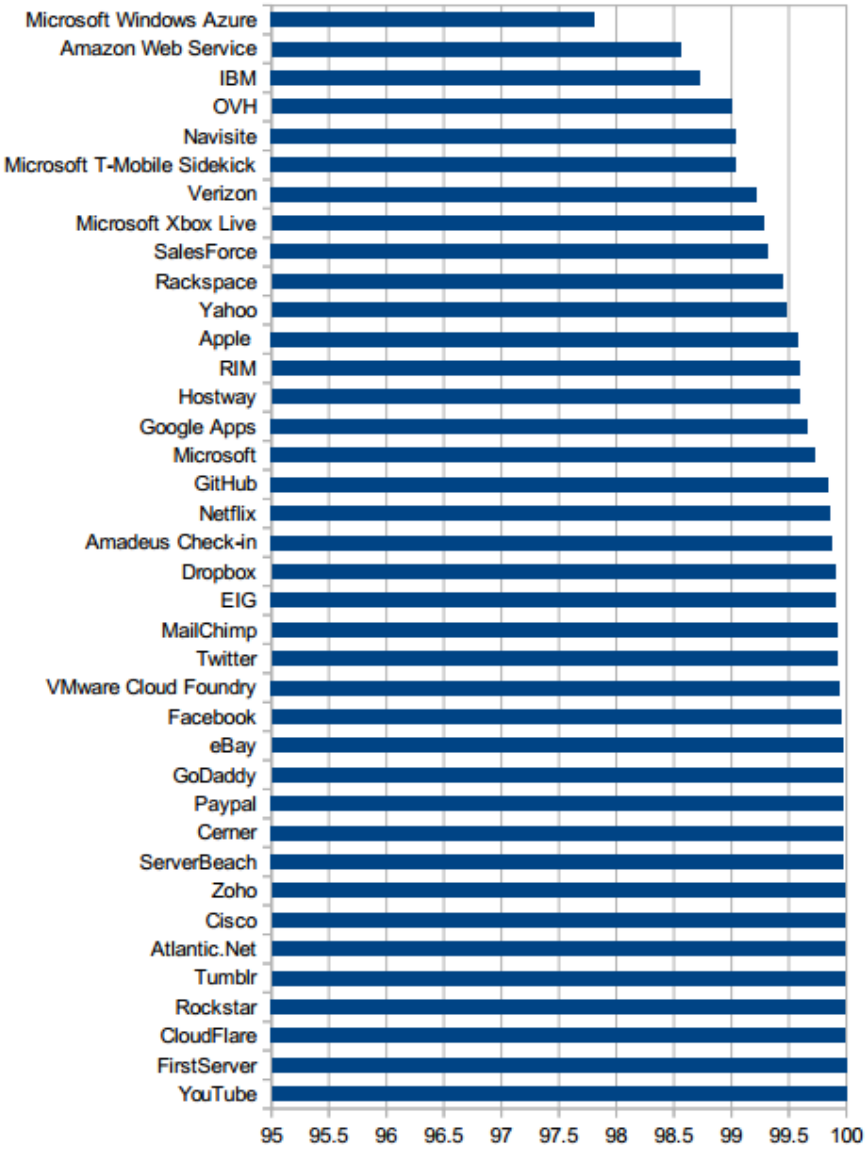
Over the past couple of years accessibility, one of the most important properties of cloud computing, has failed to demonstrate its expected level. The cloud is branded as accessible from any point and at any time and that precisely is the reason many companies decided to adopt it. However, the reality is somewhat different. From outages to data loss, the cloud computing industry has experienced all sorts of issues [10]. "Ironically, existing SaaS products have set a high standard in this regard. Google Search is effectively the dial tone of the Internet: if people went to Google for search and it wasn't available, they would think the Internet was down. Users expect similar availability from new services, which is hard to do." In order to raise awareness of this problem, the International Working Group on Cloud Computing Resiliency (IWGCR) started monitoring these failures in 2012, and that practice was continued within the work for this Master Thesis. The information is published on the website of IWGCR (www.iwgcr.org) and then summarized into semi-annual reports.



Picture 1.1 A Representation of Information Collecting by IWGCR

Picture 1.1 shows the way information about failures is collected. The majority of information is gathered from the press reports found on the Internet. The second source is the RSS feed provided by the providers themselves. Some of the providers run status pages setup to inform users about failures and keep them updated on their progress with resolving such failures. Other providers, however, often try to cover up the issues, which leads to dissatisfaction of the users, who then usually express it on the social networks, thus making them the third source of information for the website. Lastly, some of the information is obtained from case studies.

The reports provide statistics on the availability and a ranking list of 38 providers.



Picture 1.2 Ranking of the Cloud Computing Providers

Picture 1.2 shows the ranking list of cloud computing providers based on availability. It can be seen that some of the biggest providers, such as Amazon and Azure, have demonstrated relatively poor performance.

The report also lists some of the biggest failures for each of the year it's monitored the cloud computing industry.

1.5. Statistics

The statistics in Table 1.1 show that on average the providers do not perform in the way they are legally obliged by the service-level agreements (SLA). The majority of providers guarantee an availability of 99.95% per year.

Table 1.1 Downtime Statistics of the Cloud Computing Providers [11]

Minimum	0.17 h
Maximum	383.54 h
Total	2595.75 h
Average annual downtime	370.822 h
Average annual availability	85.54%

While it is obvious that this approach of measuring availability has many drawbacks, it is clear that these figures can only be underestimated. However, it was the main motivation for the project Resilience and this Master Thesis. All of the models with a more of an "engineering approach" were built on top of this approach. In addition, since this ranking list was based on the availability, some providers are highly ranked even though they experienced data loss, which severely affects their reputation.

2. Evaluation of the Cloud Computing Providers

There is no universally accepted metric of evaluation of cloud computing providers. And there will likely never be one. The evaluation of cloud performance depends so much of the needs of the customers, and at the same time there are so many criteria to compare the cloud that the possibilities of choosing a single figure of merit are endless. This chapter will show some of the possible scenarios and explain how the evaluation of the cloud computing providers' performance was measured for the purposes of this Thesis.

2.1. Defining Cloud Performance

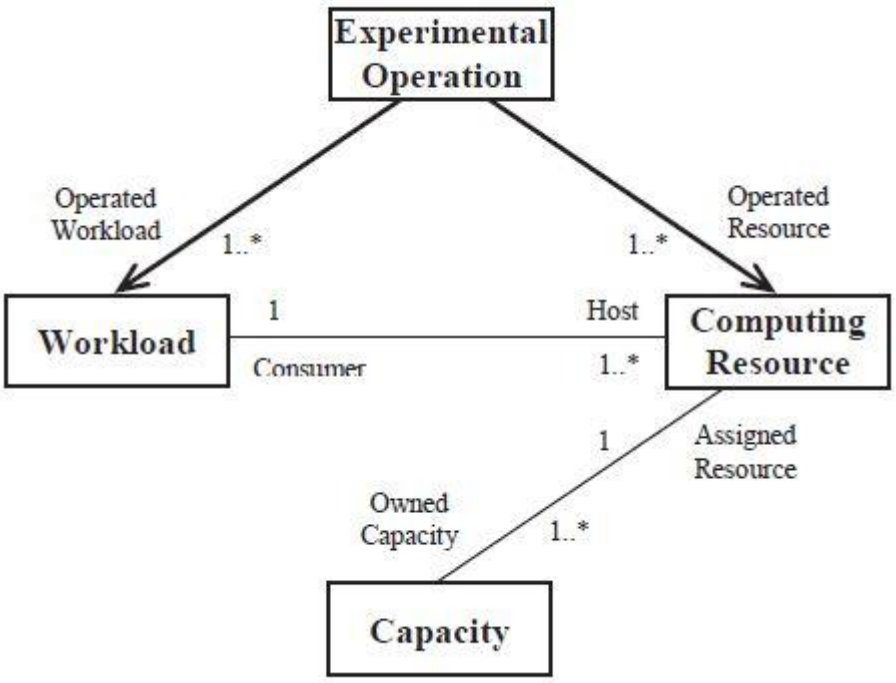


Figure 2.1 Conceptual model of Cloud services performance [12]

Figure 2.1 shows how a possible evaluation scenario can be created. The scheme is most conveniently summarized by the following rule:

Performance evaluation: evaluating ***[capacity]*** of particular ***[computing resources]*** with particular ***[workload]*** driven by a set of ***[experimental operations]***.

Physical property part:

- Communication
- Computation
- Memory (Cache)
- Storage

Capacity:

- Transaction Speed
- Latency (Time)
- Availability
- Reliability
- Data Throughput (Bandwidth)
- Variability
- Scalability

Experimental Setup Scenes:

- Single Cloud Provider vs. Multiple Cloud Providers
- Single Cloud Services vs. Multiple Cloud Services
- Single Service Instance vs. Multiple Services Instances
- Single Instance Types vs. Multiple Instance Types
- Cloud Exclusive vs. Client Cloud

Operational Scenes:

- Repeating Experiment for a Period of Time
- Repeating Experiment for a Number of Times
- Different Physical Deployments of Cloud Resource
- Different Geographical Locations of Cloud Service
- Different Geographical Locations of Client
- Sequential Workload
- Concurrent Workload
- Shared Workload between Cloud and In-House Resources
- Increasing Amount of Cloud Resource with the same Amount of Workload
- Increasing Amount of Workload with the Same Amount of Cloud Resource
- Increasing Amount of Workload with Increasing Amount of Cloud Resource
- Different Providers with the same Amount of Workload

- Different Sources (Same Provider) with the Same Amount of Workload
- Different Instance Types (Same Service) with the Same Amount of Workload

An example of an evaluation scenario would be evaluating computation speed of Amazon Elastic Compute Cloud (EC2) by repeating a computation of square root of 8 for a number of times. Also, one computing resource can have more owned capacities; for example, variability can be added to the evaluation of the computation speed.

It is clear that it these possibilities can hardly ever be all tested, so for the purposes of this Thesis, the following scenario was chosen and tested:

Evaluating computation availability of various cloud computing providers by repeating a test for a period of time.

In this case, it is important to define availability and the test that was going to be performed.

2.2. Defining Availability and Failures in Cloud Computing

Availability and failures in cloud computing are slightly different than what they are otherwise considered to be. The standard definition of availability is:

"A measure of the degree to which an item or system is in an operable and committable state at the start of a mission, when the mission is called for at an unknown (random) time" [13].

However, it is hard to define the "operable and committable state" of a cloud computing provider. First of all, it is because of the uncertainty of the culpability for the failure. Often times, a cut in the network turns the cloud unreachable. Secondly, if a single instance is not working properly, it is hard to prove it and many providers only take into consideration the cases where multiple zones were affected with unavailability [14]. However, those two might not even be the biggest problems. Looking at the Service Level Agreements (SLA), there is a lot of talk about connectivity and uptime, but very little mention of the cases where virtual machines become slow. On the other hand, the data collected from press reports and social networks reveal many customers complaining about the very issue. Usually, customers can successfully connect to their machines but the machines are

unusable because of its slow performance. An example of the case is Google's Gmail service that experienced increased latency in January 2014.

However, it is hard to decide at which point the machine's tardiness has become an outage, a failure. For example, 57 percent of online shoppers will wait three seconds or less before abandoning the site [15]. So, even if the page eventually loads, the cost of the customer loss due to the delay may be extremely high.

3. Real-time Tracking of the Availability of Cloud Computing Providers

Given the importance of the availability for cloud evaluation and due to the shortcomings of the previous approach, a new way of availability tracking has been implemented. In order to understand the concept, it is important to understand the environment in which it was built. Therefore, the next three subchapters will explain the use of ViFiB (a decentralized cloud solution), SlapOS (software that manages ViFiB) and Re6st (a dynamic routing protocol).

3.1. Decentralized Cloud Solution ViFiB

Recently, many studies have shown that the cloud's centralization is a threat for the data safety. Martin Fink, Hewlett Packard's Chief Technology Officer (CTO), Director of HP Labs and General Manager of HP's Cloud business unit, on a general session in June 2014 presented the model of „After Cloud" that is planned to be developed and fully enabled for use in 2018. He criticized the today's cloud for the fact that with so much compute and storage power put in one place, the cloud has to take many different roles, such as: translator, orchestrator, arbitrator, coordinator, aggregator, replicator, anonymizer, board guard, learning engine, etc. And that means that there is too much data and too many legal and privacy issues associated with a single central point [16].

University of California Berkley's most important paper about the cloud quotes: "Just as large Internet service providers use multiple network providers so that failure by a single company will not take them off the air, we believe the only plausible solution to very high availability is multiple Cloud Computing providers. The high-availability computing community has long followed the mantra "no single source of failure," yet the management of a Cloud Computing service by a single company is in fact a single point of failure" [10]. It also mentions the fact that having multiple datacenters in different geographic regions and different network providers does not guarantee the resilience since they may share common software infrastructure and accounting systems, as well as the possibility of the provider running out of business.

Given all the criticism cloud has received, as well as the research conducted on the downtime statistics, the work of this Thesis is based on a cloud model created by one of the partners of the International Working Group on Cloud Computing Resiliency, Nexedi. Nexedi created, ViFiB, a decentralized cloud solution. Instead of storing data in enormous datacenters, ViFiB proposes a solution that distributes the datacenter to many small servers run in homes around the world.

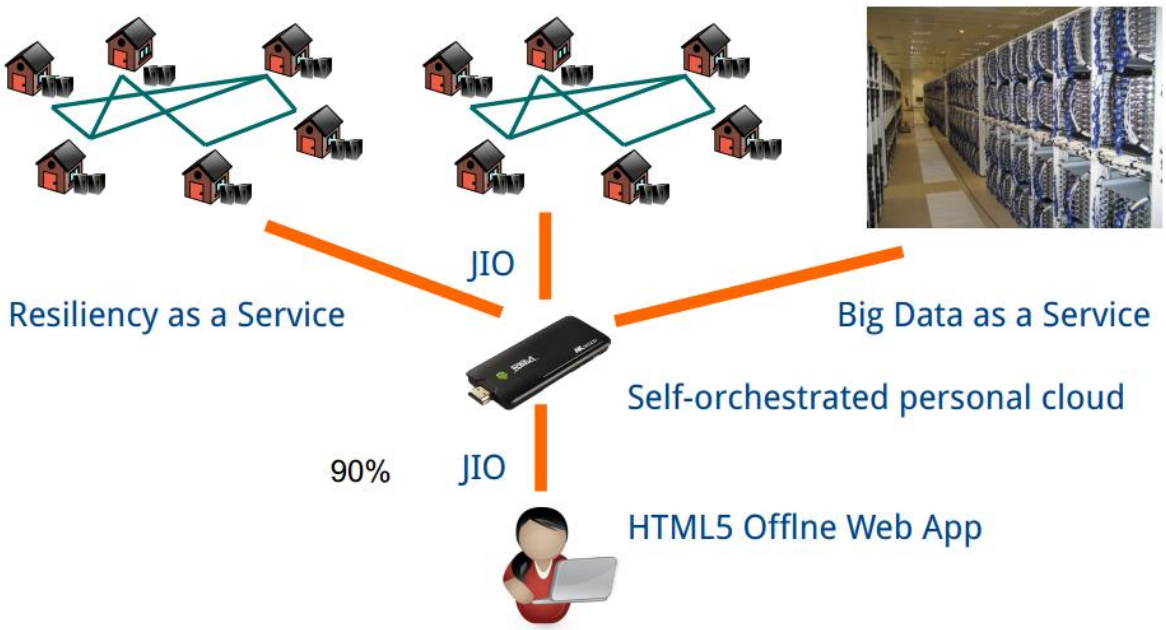


Figure 3.1 The Architecture of ViFiB

ViFiB leverages existing hardware to store data across the globe. The information of a single customer is encrypted and stored on three different locations to ensure resiliency, while the customer uses the server that is the closest to his location to diminish latency and transfer cost. In case of a disaster or downtime of a server, the data is replicated and the cloud continues to operate. ViFiB's customers use "bare metal", there is no virtualization. The ViFiB network is volatile, so the information changes locations periodically, leaving no place for attacks on specific customer's data.

3.2. Simple Language for Accounting and Provisioning operating system

Simple Language for Accounting and Provisioning operating system (SlapOS) is the core of ViFiB. It is a multicloud orchestrator that manages the whole infrastructure of ViFiB. It can automate the deployment and configuration of applications in a heterogeneous environment. SlapOS supports IaaS, PaaS and SaaS applications. Other than managing the infrastructure, SlapOS also handles the provisioning of application instances with accounting and billing modules. Even though SlapOS is made to fit the needs of ViFiB, it can manage instances of other cloud computing providers in the same modules. All it takes to set it up is to install SlapOS on each of the machines (Figure 3.2).

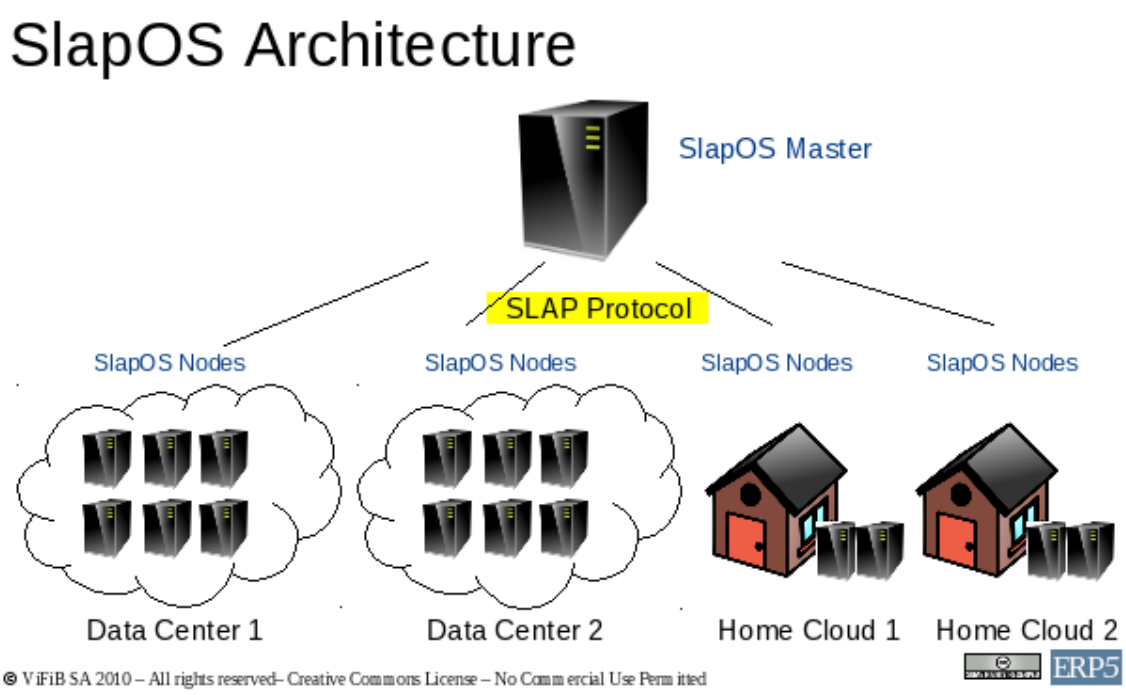


Figure 3.2 SlapOS [17]

The main idea behind SlapOS is that rather than trying to build multiple reliable data centers with redundant network, redundant power supply, redundant air conditioning, redundant staff, redundant space, etc. in order to reach 99.999% availability, SlapOS combines unreliable hosting (home cloud, low cost dedicated servers) and by doing so reaches the availability of 99.999%.

By selecting independent sources, SlapOS can survive any case of *force majeure*: strike, earthquake, political decision, electricity breakout, etc. That said, SlapOS can also be used to operate traditional data centers more efficiently.

SlapOS is made to considers “everything is a process” and that a system is made of processes which exchange information through network services based on various protocols: HTTP; memcached, mysql, etc. That being the case, SlapOS is built with the belief that data replication or energy savings should be implemented by the application itself rather than by the infrastructure. By taking into account usage or data specificities, optimal policies for redundancy and energy savings can be implemented.

SlapOS is easy to use and anyone can become a cloud provider by using SlapOS.

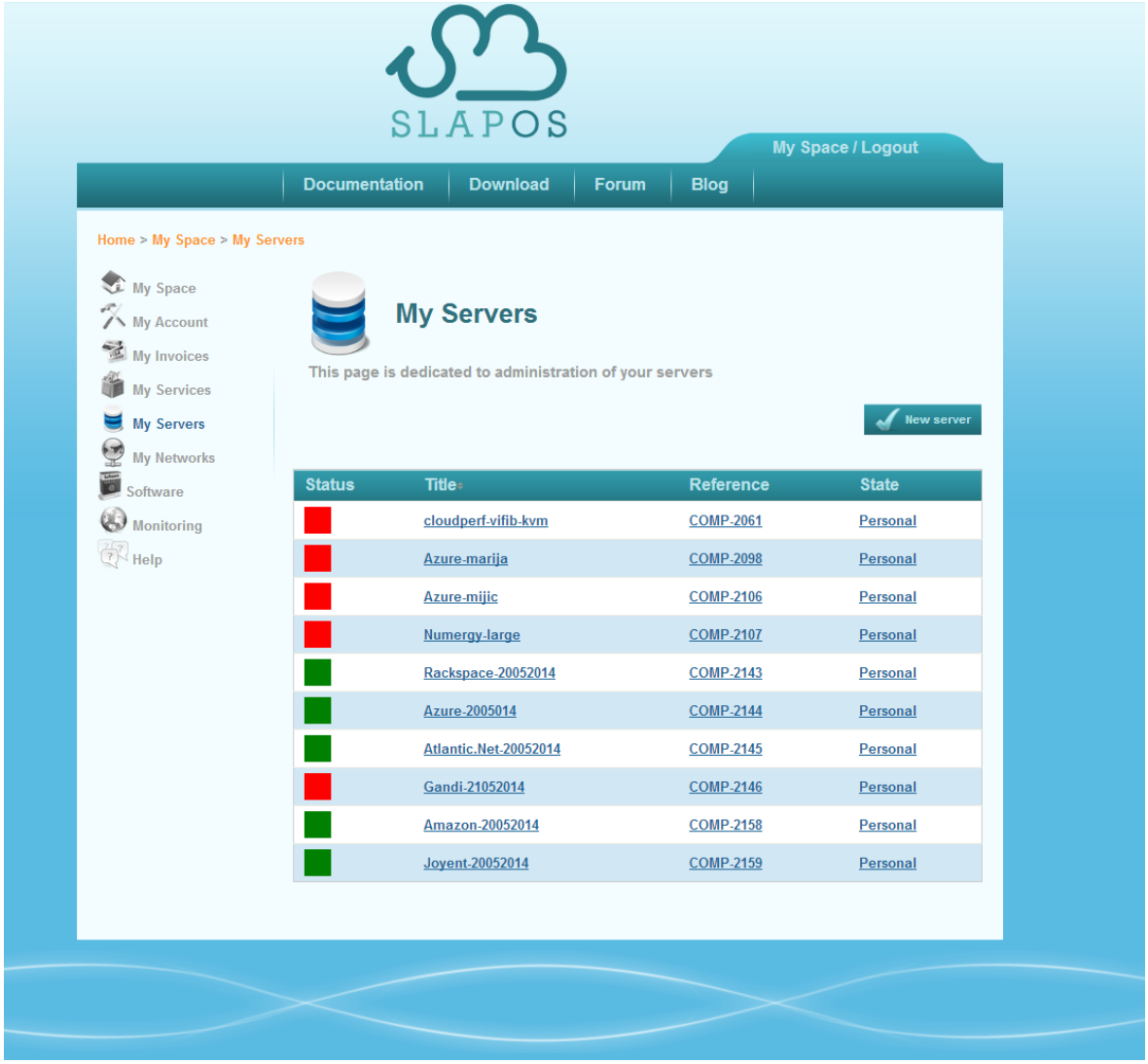


Figure 3.3 The Interface of SlapOS [18]

Figure 3.3 shows the interface of SlapOS. Both ViFiB machines and other providers' machines are provisioned in the same way. A red square represents a machine that is not running and a green one means the machine is up and running. The process of installing is finished within minutes and the provisioning is fully automated. The only prerequisite is that all of the users use IPv6. In the next subchapter it will be explained how the lack of IPv6 protocol can be overcome.

3.3. Re6st Resilient Overlay Networking System

Re6st optimized routing technology shares similarities with peer-to-peer (P2P) technologies such as Skype, PPStream or BitTorrent. re6st initializes a mesh of direct routes between selected edges in a network. It leverages the mesh to create indirect routes between all edges. Thanks to babel (RFC 6126) distance vector routing protocol, the best route is always selected to interconnect edges [19].

By using Re6st, customers do not have to change the existing IPv4 infrastructure or applications. Its automated IPv6 address allocation makes it suitable for scaling up to very large networks and guarantees interoperability with other organizations.

3.4. Availability Evaluation Model

So far, there is no way a non-profit institution could track the availability of the cloud computing providers and, by doing so, achieve user and data protection. A new model for availability, presented at The Intersection of Cloud and Mobility conference at National Institute for Standards and Technology (NIST) is a part of the research conducted for this Thesis and will be described in this subchapter.

First of all, every provider should be more transparent about outages and should run a status website describing in detail what caused the issues. Some of the major providers already have similar websites, but they are often not updated until there is an outburst of angry customers on social networks and forums. In addition, most of the providers only keep records of the outages up until two weeks before the date of the access of the website. That is insufficient data to evaluate the overall performance of the provider.

In order to verify the records on the status websites, there should be a non-profit organization running the availability test at all times. One of the ways to achieve that, as used in this Thesis, is by using SlapOS.

SlapOS architecture uses two types of servers: SlapOS Master and SlapOS Nodes. SlapOS Nodes are hosted in multiple data centers or at home. SlapOS Nodes exchange information with the SlapOS Master through the SLAP protocol. SLAP stands for "Simple Language for Accounting and Provisioning". Slap Master is in charge of the SlapOS nodes and decides which software they should install and which instances of software they should run. A Node periodically 'asks' the Master what it should do and that is the principle this model bases its availability evaluation model.

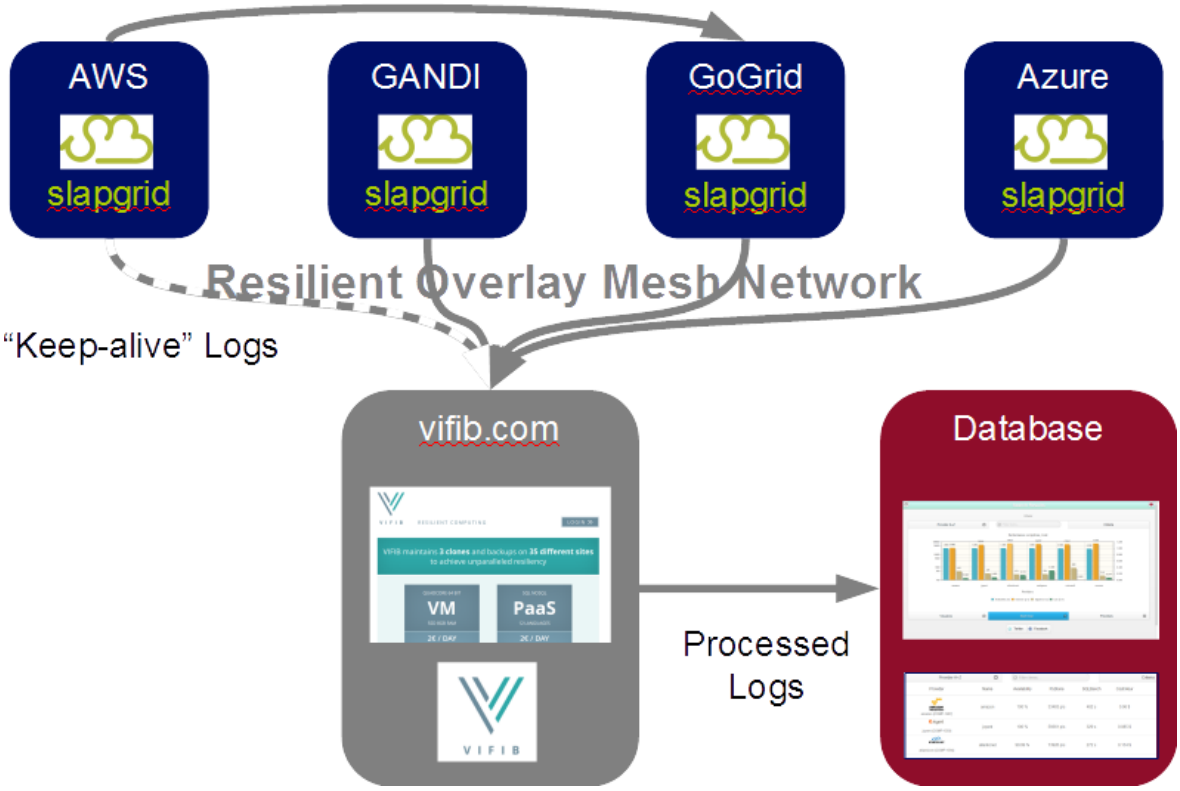


Figure 3.4 The Architecture of the Real-time Cloud Availability Tracking Model

Since the Node contacts Master every minute, the absence of a message can mean three things:

- The SlapOS system is down
- There is a network issue between the Master and the Node
- The Node (provider) is down.

The first is easy to identify, when none of the Nodes are pinging the Master, it is obvious that it is an internal problem of the Master. The second problem is eliminated by Re6st that always finds a route that can connect the Master and Node. This means that the third case can be easily identified.

Figure 3.4 shows the architecture of this model. The Nodes in reality cloud computing providers, such as Amazon, Gandi, Azure, etc. They all have installed the SlapOS Node and through SlapOS and ViFiB it is possible to have keep-live logs of the availability of the machines. The logs are processed afterwards and published to a database.

4. Experimental Evaluation of the Cloud Computing Providers and Failure Prediction

4.1. Evaluation with PyStone and SQLBench

The first part of the evaluation was conducted with the use of two widely used benchmarks: PyStone and SQLBench. The machines that were tested were acquired for this Thesis and used only for the testing. Each of the machines had SlapOS installed. Within the SlapOS environment, a service called Erp5 Test Node was installed. It is the same service in charge of the availability measuring, but was this time for the benchmarks that it runs on the machines. The results of the benchmarks are sent to the SlapOS Master that stores them into a database, as shown on

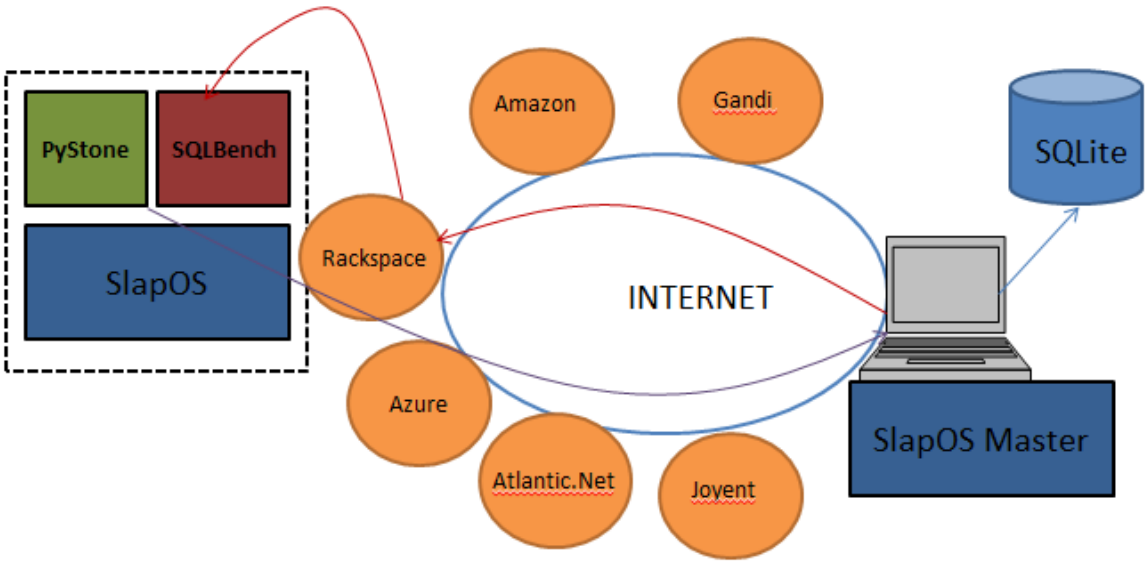


Figure 4.1.

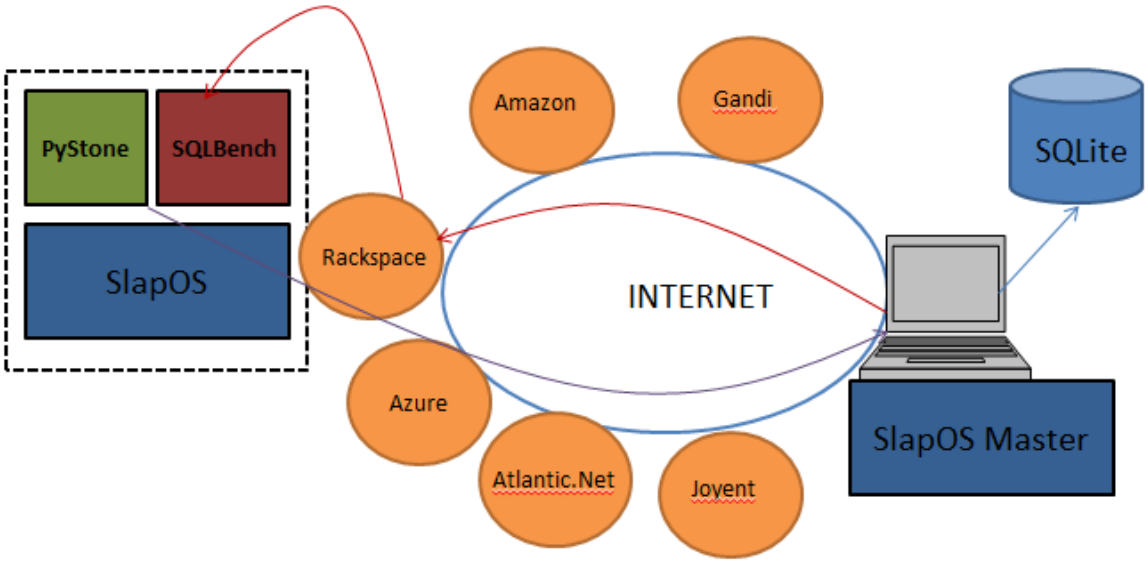


Figure 4.1 The Cloud Computing Providers Evaluation Model

The following table contains information about the machines' specifications:

Table 4.1 Specifications of the Virtual Machines Used

Cloud provider	Instance name	CPU	RAM	Disk
Amazon	m1.small	1	1.7 GB	160 GB
Rackspace	2GB Performance	2	2 GB	40GB SSD system disk+20GB SSD data disk
Azure	Medium (A2)	2	3.5 GB	
Atlantic.net	Medium	1	2 GB	80 GB
Joyent	g3-standard-1.75-kvm	1	1.75 GB	56 GB
Gandi	Large	2	2 GB	50 GB

The machines are in a range of 1-2 CPU's and have no more than 3.5 GB of RAM. The disk size varies more, but in this case it does not matter.

Table 4.2 Results of the Evaluation

	Availability	PyStones/second	SQLBench (seconds)
Amazon	100 %	167627	482
Azure	99.28%	63643	284
Atlantic.Net	99.99%	73926	272
Joyent	100%	617284	329
Gandi	100%	66385	239
Raclspace	99.7%	66452	284,5

Table 4.2 shows the results of the real-time availability tracking and the benchmarks intalled on the virtual machines. It is important to notice that the results are relevant only for the machines that were used in this testing and not the whole provider infrastructure.

4.2. Failure Prediction Model

As mentioned in chapter 3, a failure cannot be universally defined in the case where the machine is running, but it is so slow that it becomes unusable. To determine when the machine has a failure, the type of machine, its specifications and its use must be known. In this scenario, the failure of a machine will be defined as the state of the machine in which average CPU load has surpassed the number of CPU's. The operating system (OS) that was used is Linux, so all of the terms defined in the next subchapter will refer to the same OS.

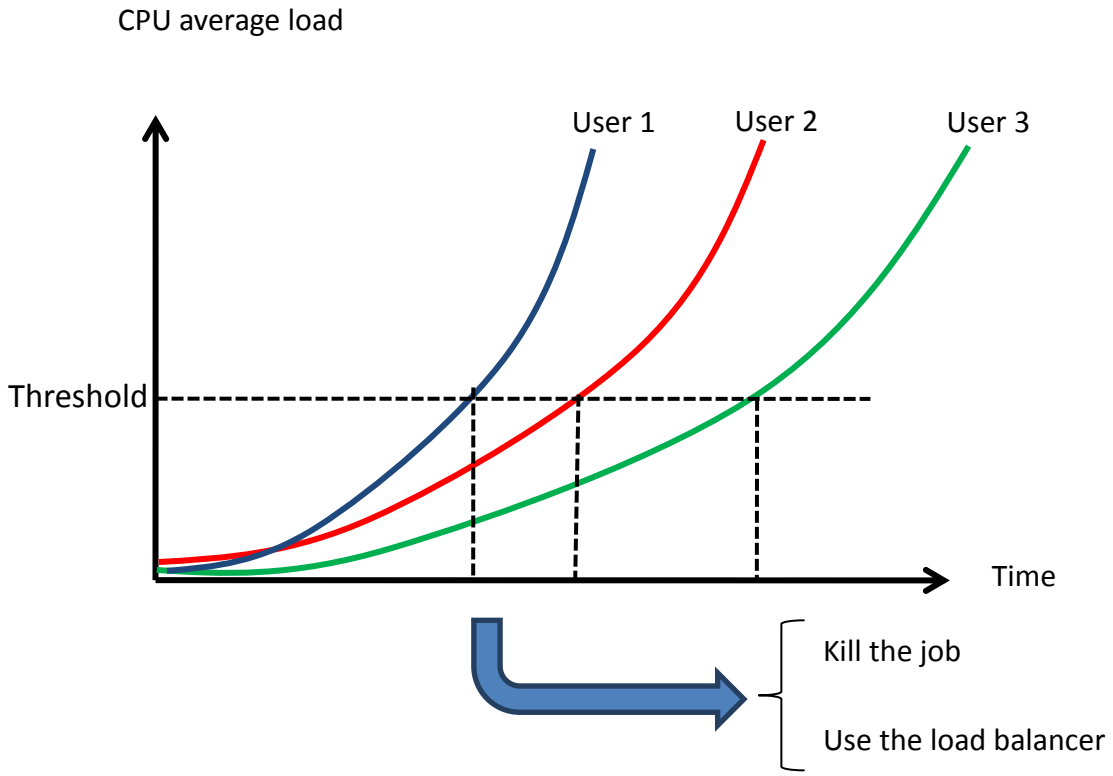


Figure 4.2 The Model of Machine Testing

Figure 4.2 shows the model of the machine testing conducted to obtain information that can help predict the failure of a virtual machine in the case of CPU overload. For a certain amount of time, a machine is stress tested. Stress testing refers to the testing of software or hardware to determine whether its performance is satisfactory under any extreme and unfavorable conditions, which may occur as a result of heavy network traffic, process loading, underclocking, overclocking and maximum requests for resource utilization. In Linux systems, stress is a tool to impose load on and stress test systems. In this scenario, the stress tool was used to impose load on disk input/output (I/O) in order to measure the growth of CPU load. Stressing I/O in practice forks n processes that call the `sync()` function in a loop. `Sync()` flushes any data buffered in memory to disk. Overloading disk was chosen because it is very difficult to saturate the machine with another technique without perturbing the measures.

Analyzing the growth rate of CPU average load can help determine if the process is going to keep growing and surpass the threshold set to be the value when the average CPU load has surpassed the number of CPU's. In other words, it allows determining if a growth rate of a process CPU average load will lead to the machine failure. The growth rates calculated for this work are based on seconds and calculated using the following formula:

$$Growth\ Rate = \left(\left(\frac{Start\ Average\ CPU\ Load}{Eng\ Average\ CPU\ Load} \right)^{\frac{1}{End\ Time - Start\ Time}} \right) - 1$$

4.2.1. Central Processing Unit Average Load vs. Usage

CPU load is a measure of the amount of computational work that a computer performs. It is calculated as an average load over a period of time, usually one, five and fifteen minutes. For example, an average load of "1.62 0.8 3.2" on a system with one CPU can be interpreted as following:

- During the last minute, the system was 62% on average overloaded
- During the last five minutes the system was idling for 20% of the time on average
- During the last fifteen minutes, the system was 220% percent overloaded on average.

The average load can also be interpreted in a way that shows how many processes had to wait for the CPU on average. In the first case, 0.62 processes would be "in the queue".

CPU usage (or process time) is measure as the percentage of the CPU's capacity used. It is the total amount of time a process has actively used a CPU since some arbitrary event. The CPU time is measured in clock ticks or seconds [20].

Studies have shown that CPU load provides a better insight into the machine state than the CPU usage. The reason CPU queue length did better is probably because when a host is heavily loaded, its CPU utilization is likely to be close to 100% and it is unable to reflect the exact load level of the utilization. In contrast, CPU queue lengths can directly reflect the amount of load on a CPU. As an example, two systems, one with 3 and the other with 6 processes in the queue, are both very likely to have utilizations close to 100% although they obviously differ [21]. This is the reason this work will only consider CPU average load in the experimental evaluation of the providers.

4.2.2. Amazon

The first machine that was analyzed was an Amazon EC2 instance. This machine has one CPU and 1.7 GB of RAM.

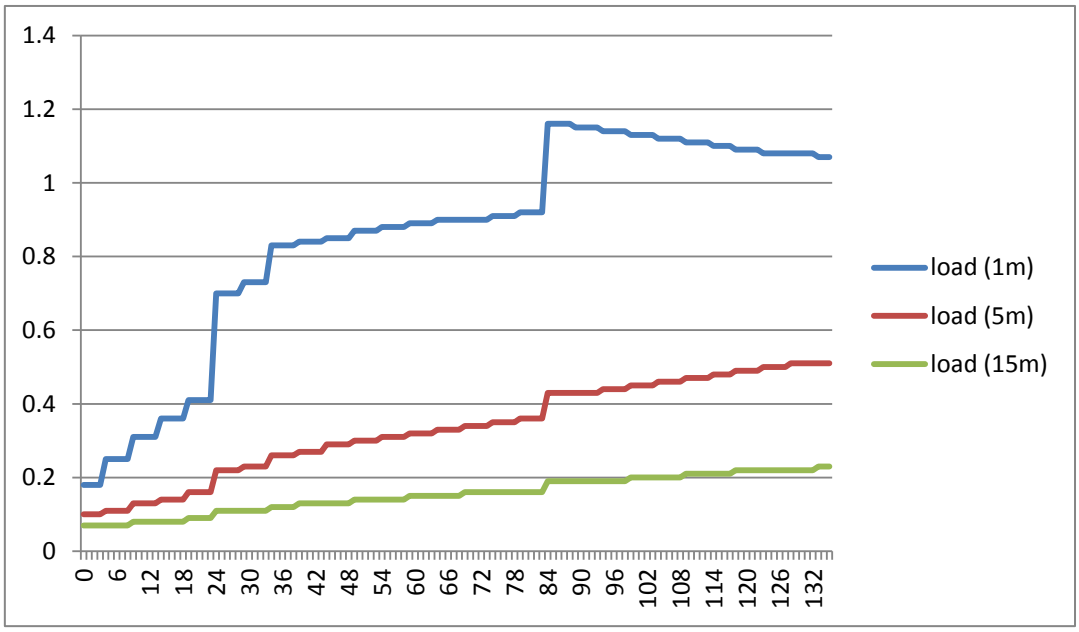


Figure 4.3 Amazon 1 process

Figure 4.3 shows the results of stress testing the Amazon machine with 1 process that `sync()` function in a loop. The load is measured for the average of 1, 5 and 15 minutes. In the case where predicting a failure is the main idea of the testing, it is obvious that the 1 minute average load will be of the greatest importance. Here, the growth is constant, but it is not consisted only of extremely high jumps in CPU average load. On the contrary there is a long period of slow growth and finally, the load starts stabilizing after it surpasses the threshold. The average growth rate calculated after the load has surpassed 70% of the threshold is 0.05822.

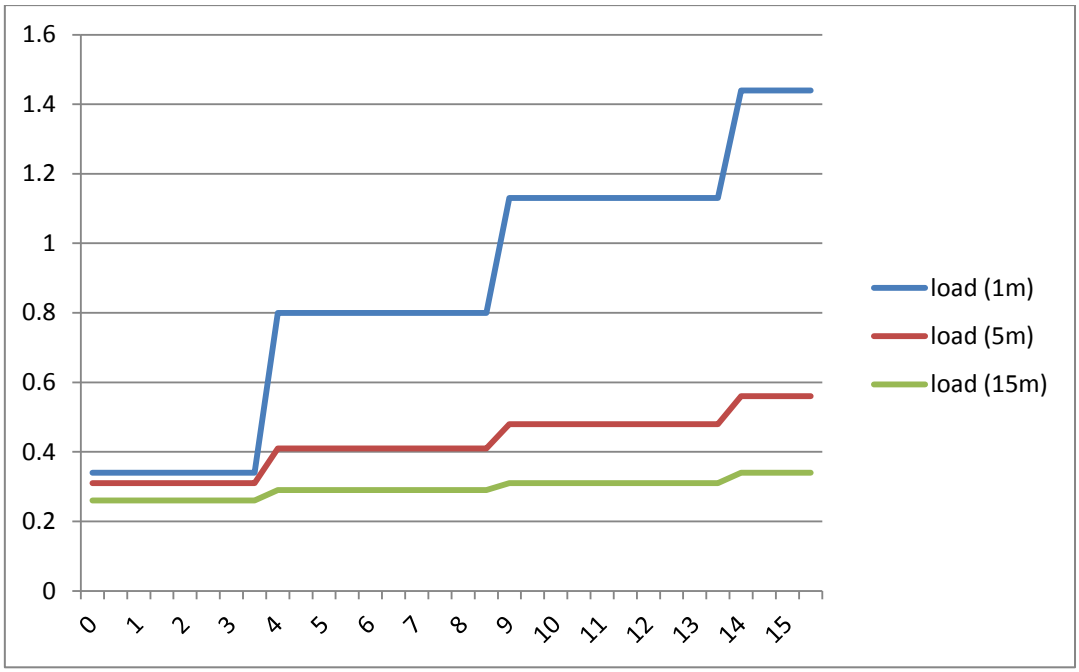


Figure 4.4 Amazon 5 processes

Figure 4.4 shows the growth of the average CPU load when the disk had 5 processes calling the `sync()` function infinitely. This scenario shows a different reaction than the first one where only one process was stressing the system. The load grew very quickly and it increased for 100 percent 3.5 seconds after the testing was started. The average growth after the load has surpassed 70% of the threshold is 0.238519.

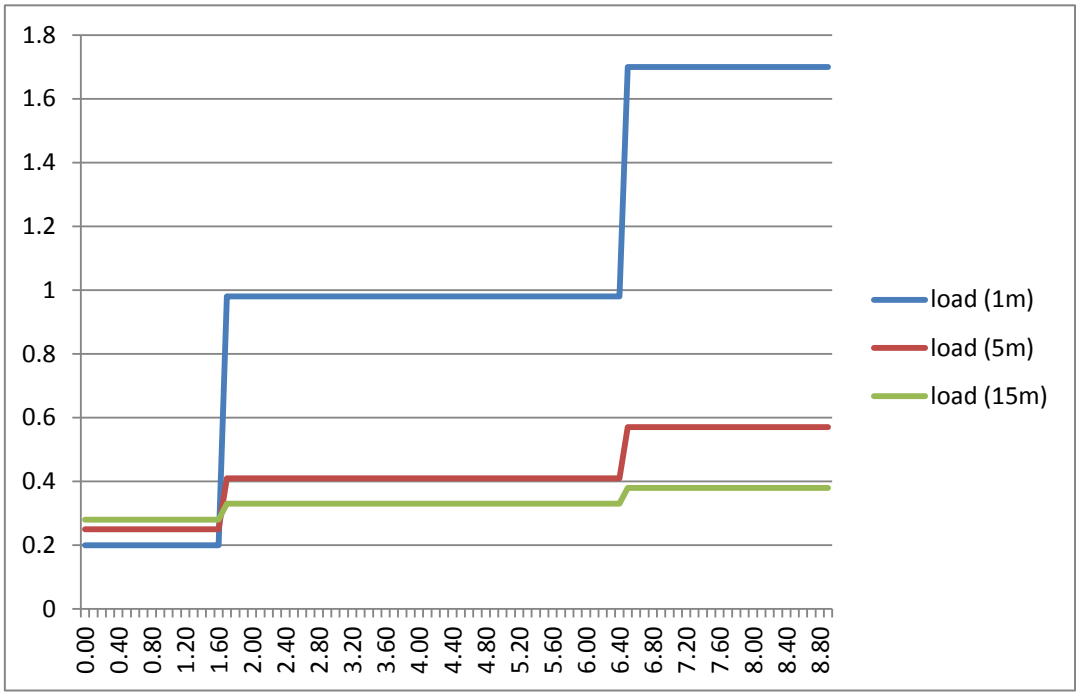


Figure 4.5- Amazon 10 processes

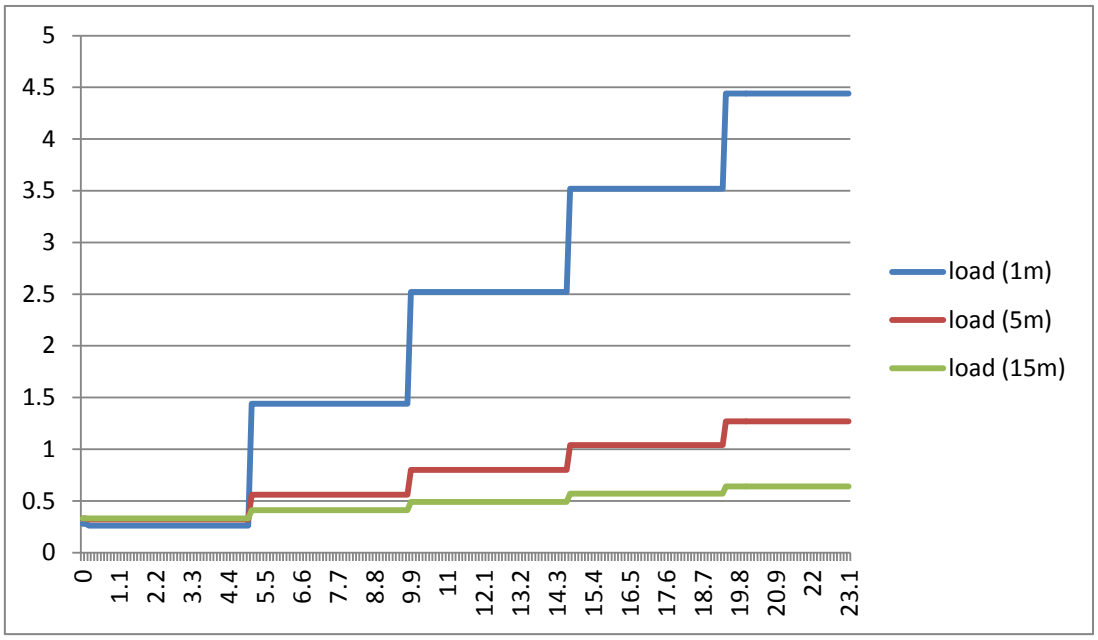


Figure 4.6 15 processes

Figure 4.5 and Figure 4.6 show processes that dramatically increase CPU average load. In addition, it takes only a couple of seconds for both of them to reach the average load of 1. The first one is a stress test with 10 and the second a stress test with 15 processes overloading the disk I/O. After the first jump in the CPU average load, it is evident that both of these processes will cause a failure of the machine and should be either terminated or moved to another, more powerful machine.

4.2.3. Joyent

The Joyent instance had 1 CPU and 1.75 GB of RAM

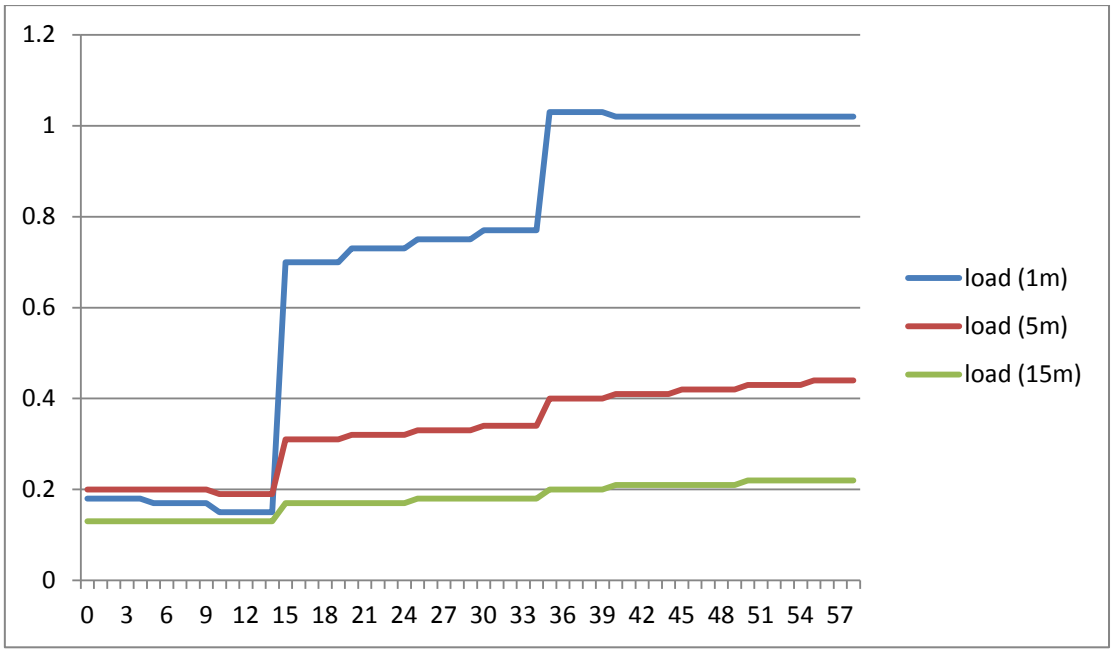


Figure 4.7 Joyent 1 process

In this scenario a Joyent machine is loaded with a stress test with only one process, but there are still big jumps, of the CPU average load. However, after surpassing the threshold, the load seems to be stabilizing. The average growth rate after the load has surpassed 70% of the threshold is 0.094767.

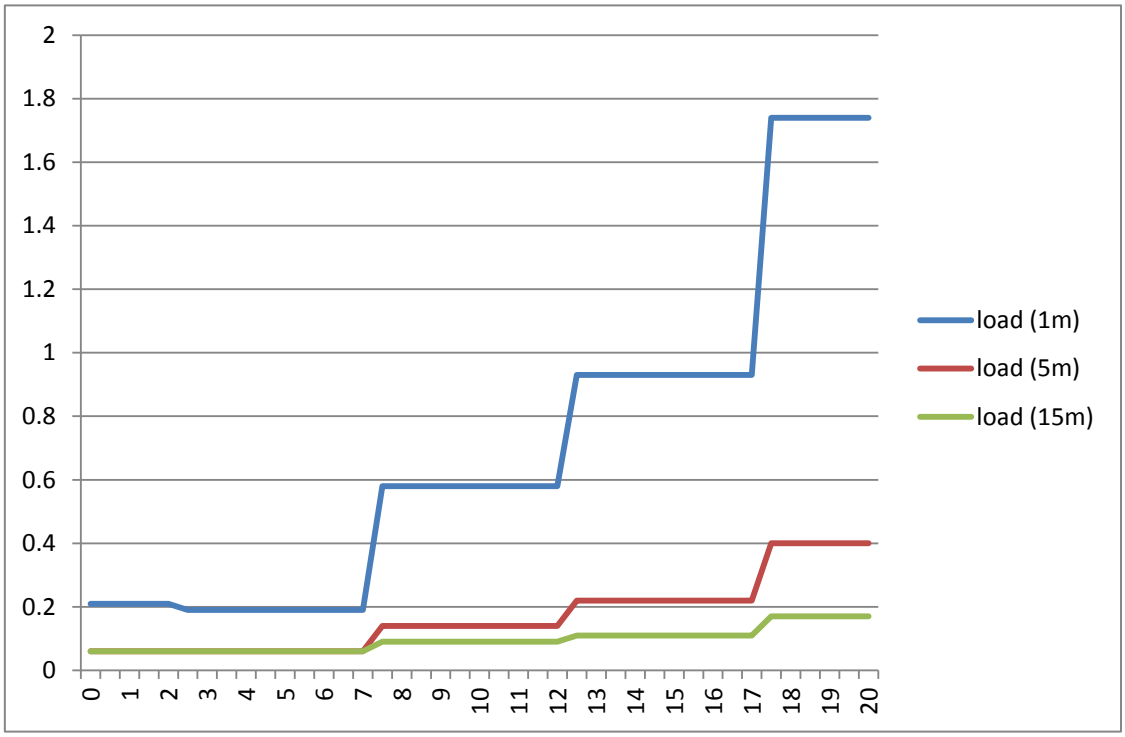


Figure 4.8 Joyent 5 processes

Figure 4.8, unlike the figure Figure 4.7 shows the growth that does not stabilize after it has surpassed the threshold. The machine is loaded with 5 processes that are calling the `sync()` function in a loop. The average growth rate after the load has surpassed 70% is 0.12642.

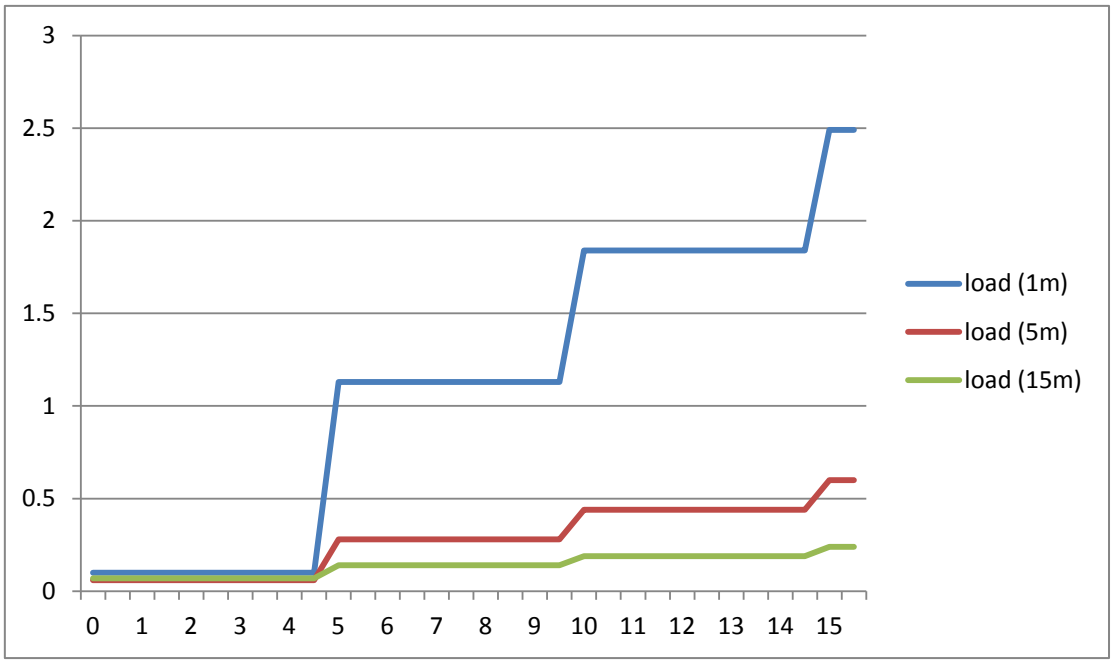


Figure 4.9 Joyent 10 processes

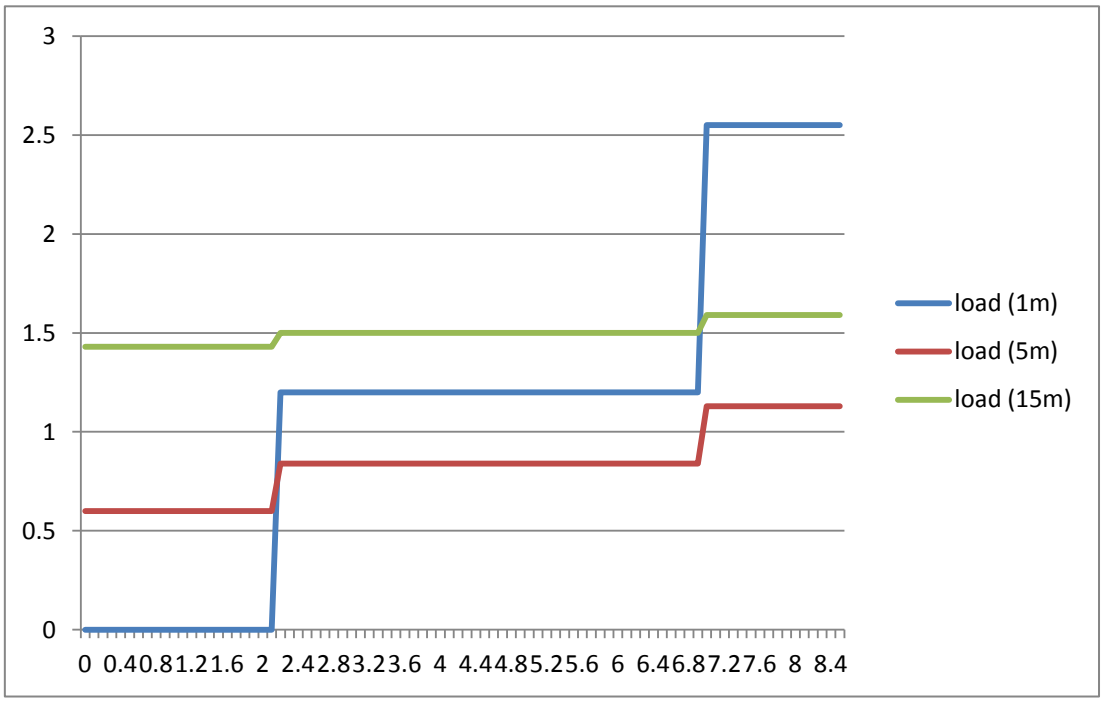


Figure 4.10 Joyent 15 processes

Figure 4.9 and Figure 4.10 show scenarios in which the average load almost immediately surpasses the threshold. These processes should be immediately terminated or moved to another machine.

4.2.4. Azure

The Azure machine had 2 CPU's and 3.5 GB of RAM.

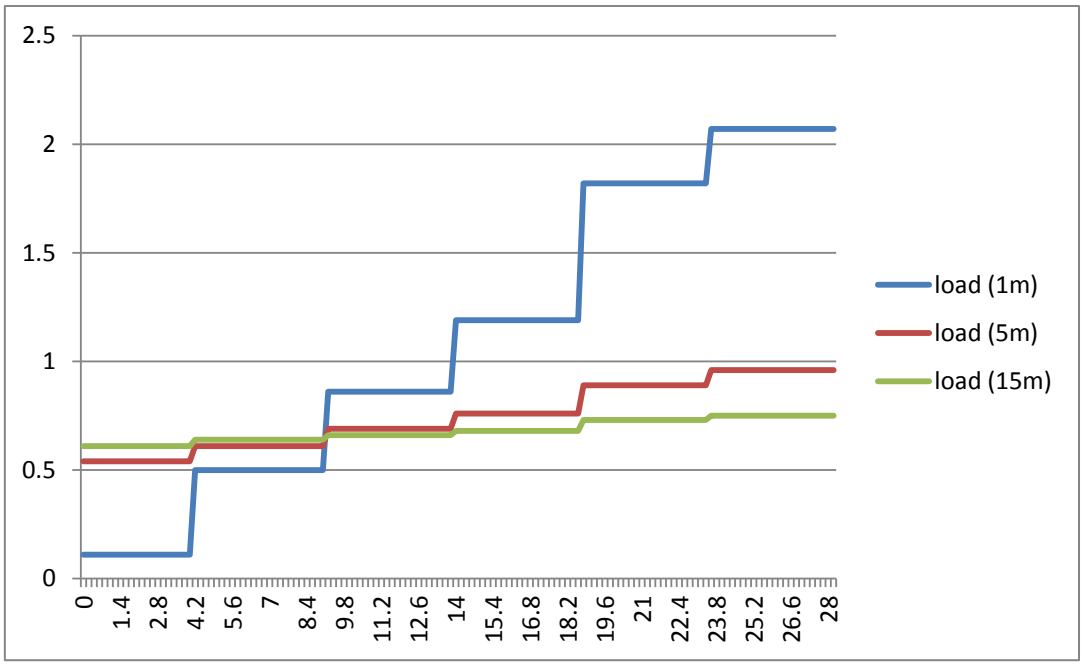


Figure 4.11 Azure 5 processes

Figure 4.11 shows the scenario in which the Azure machine was loaded with 5 processes that were calling the `sync()` function. The load is growing quickly with average growth rate of 0.18541 after the load has surpassed 50% and 0.16098 after it has surpassed 70%.

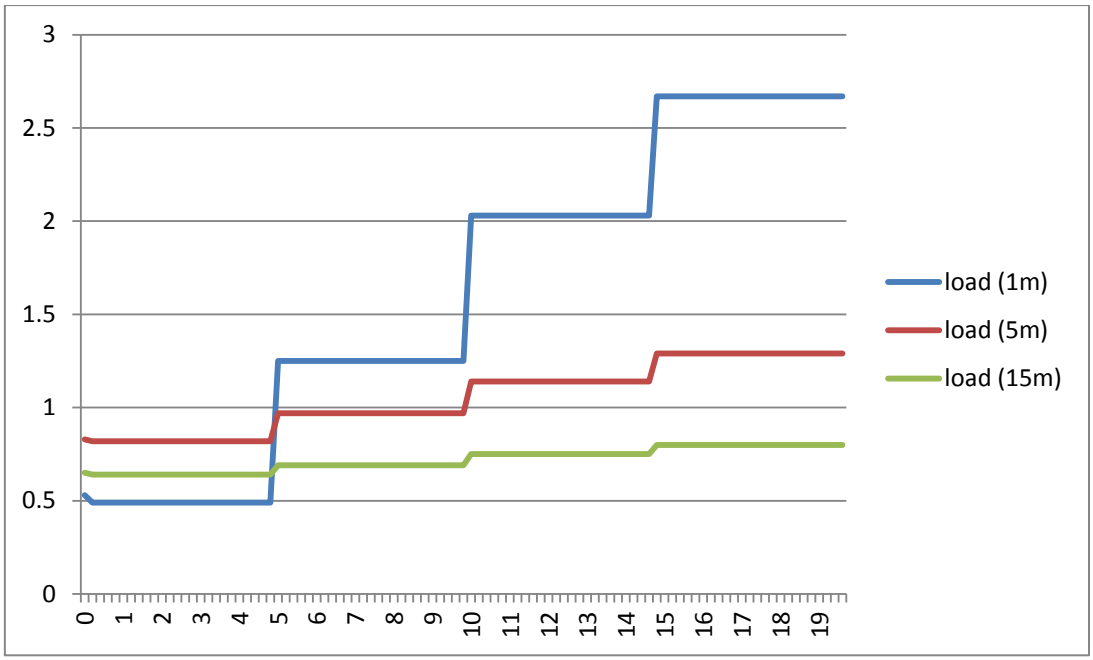


Figure 4.12 Azure 10 processes

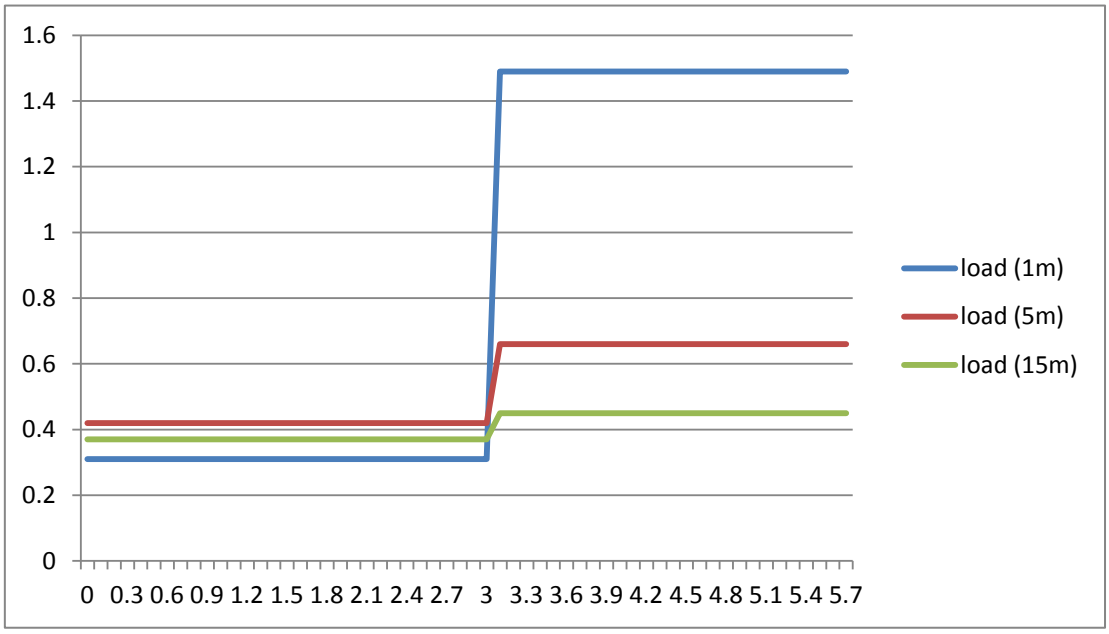


Figure 4.13 Azure 15 processes

Figure 4.12 and Figure 4.13 are without a doubt a potential cause of a failure and these processes. The load with 10 processes takes half as much time as the load with 5 processes (which was also critical) to reach the threshold and the load with 15 processes jumps from an average load of 0.35 to 1.5, which should alarm the system to kill that process.

4.2.5. Rackspace

The Rackspace instance that was stress tested has 2 CPU's and 2 GB of RAM.

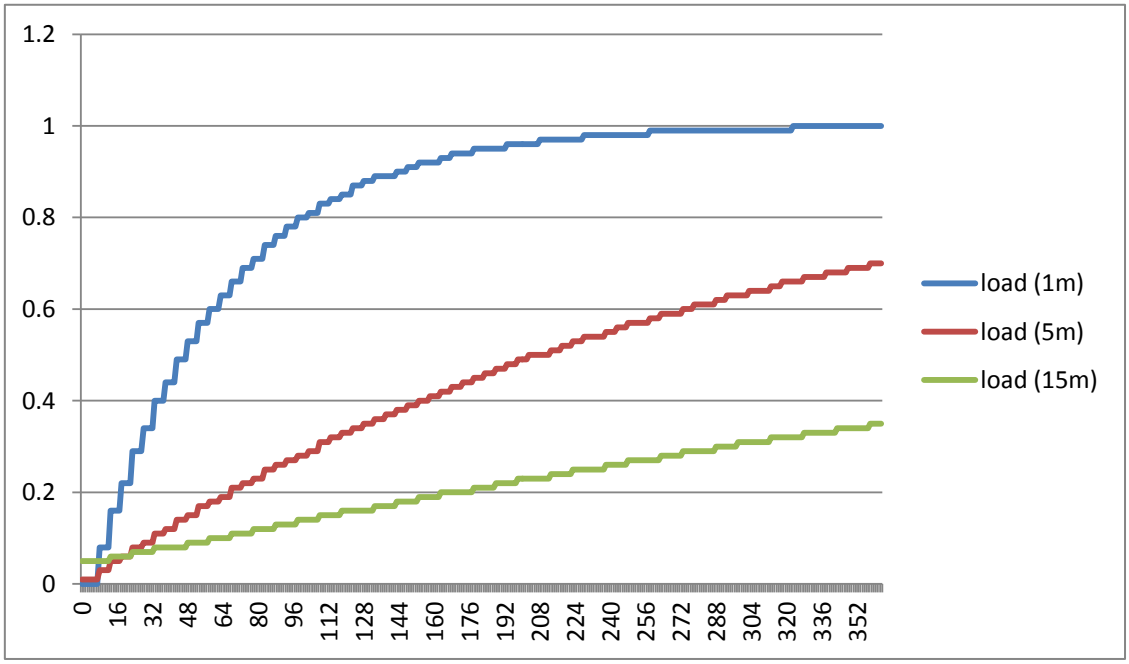


Figure 4.14 Rackspace 1 process

Here, the Rackspace virtual machine is stressed with one process calling the function. The average load is at first growing, but then the growth starts declining, in a way that resembles the logarithmic function. This is an example of a growth that should not alarm the system since the growth rate is only 0.018215 at the 50% of threshold, and more importantly the growth rate is declining.

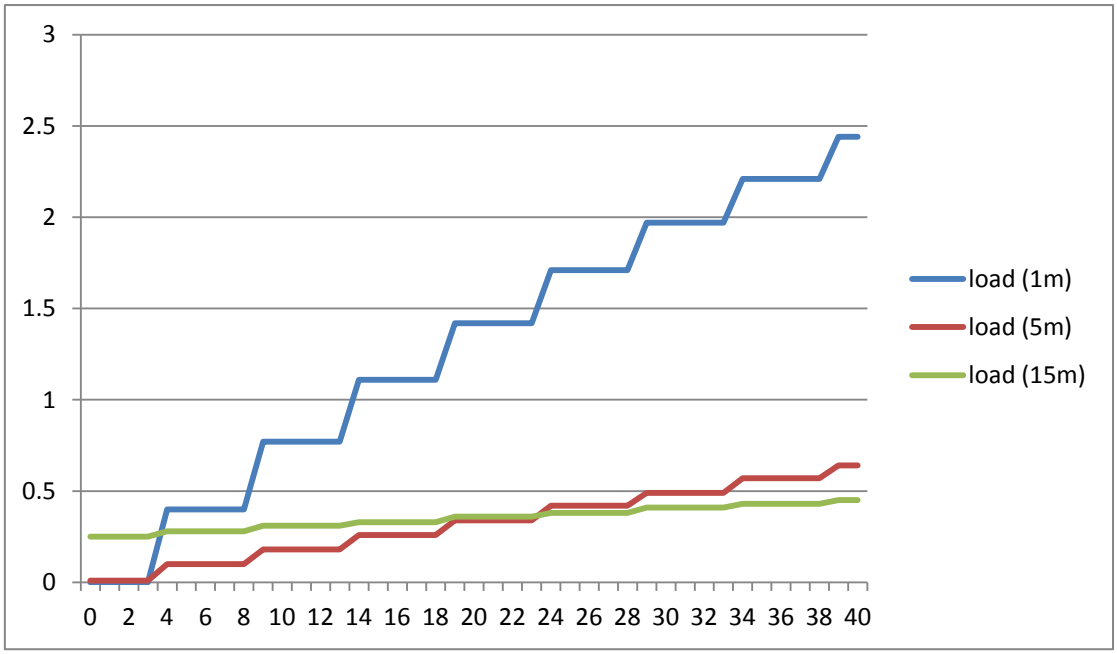


Figure 4.15 Rackspace 5 processes

Figure 4.15 shows the scenario in which the machine is loaded with five processes calling the `sync()` function. Unlike in the previous case, the load doesn't stabilize here.

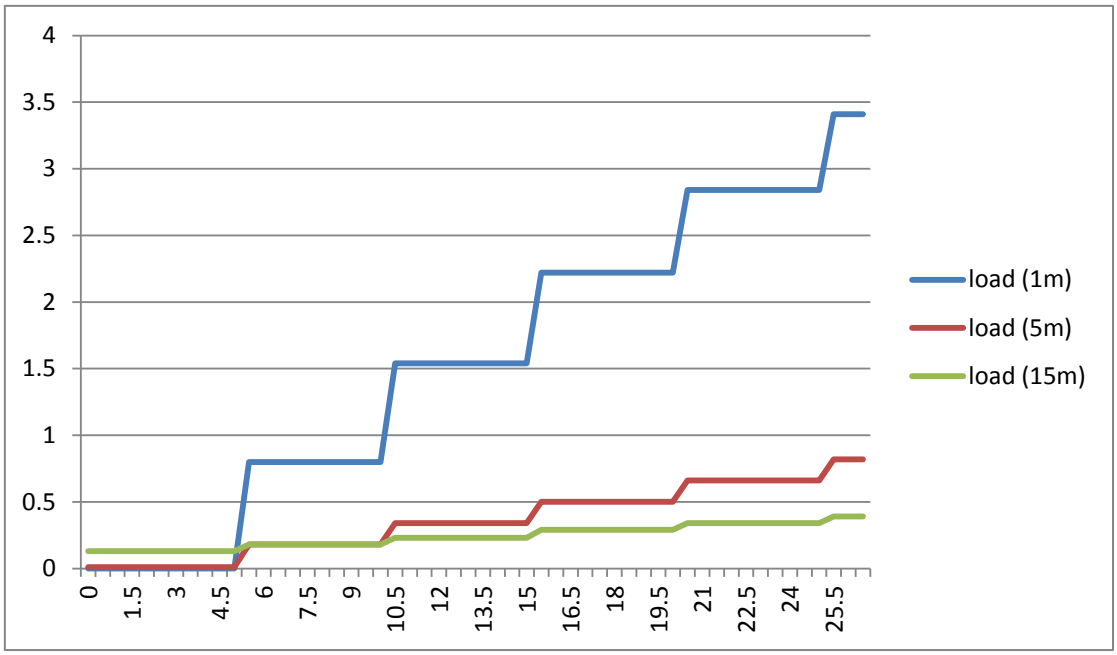


Figure 4.16 Rackspace 10 processes

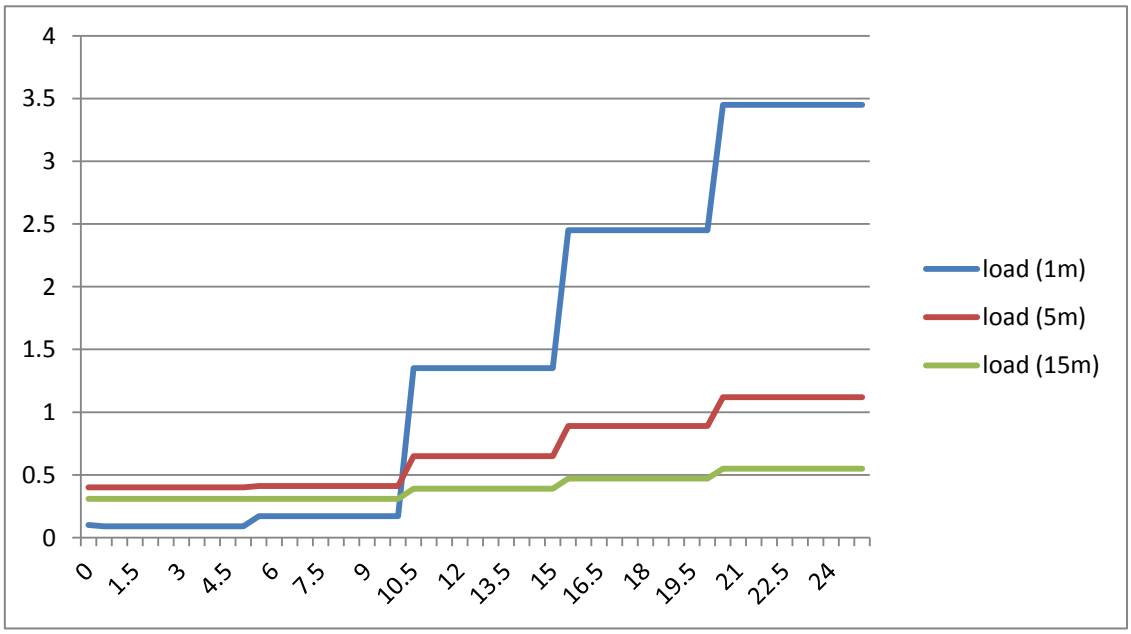


Figure 4.17 15 processes

Figure 4.16 and Figure 4.17 represent the case where the machine is loaded with 10 and 15 processes, respectively.

4.2.6. Comparison of Test Results and System Decision Making

This testing was conducted to empirically determine which processes present a potential threat to the system with the growing load they impose on CPU. However, it can also be used to compare providers of similar characteristics.

Both Amazon and Joyent have 1 CPU and approximately the same RAM size. However, in the case of the loads with 5, 10 and 15 processes Joyent takes more time to reach the threshold, giving it a certain performance advantage.

In comparison of Rackspace and Azure, both with 2 CPU's, Rackspace's load growth is slower and in the scenario with one process it fully manages to continue operating, with the average load below 50% of the threshold.

The decision on the system is based on the following set of rule:

If the average load of CPU has reached 70% of the failure threshold, calculate the average growth rate. If the average growth rate is bigger or equal to x, take actions.

Given the experiments, it is clear that loads that are less that 0.1% at the moment the load hits 70% of the threshold are the ones that system could stabilize, sometimes even after they surpassed the threshold, setting 0.1% percent average growth rate to be the x that determines whether to take actions on the process or not. Growth rate as small as 0.2% in the case of Amazon instance loaded with 5 processes, did not stabilize.

However, in order to be accurate, the average growth rate should be measured at 50% of the growth rate and then again at 70% to determine if the load growth rate is declining fast. For example, if the growth rate from 50% to 70% has dropped for more than 50%, it could be an indicator that the load will decrease and will not negatively affect the machine.

To ensure the actions are being taken on the right process (and the right user if they are sharing the CPU), a Python script was created for the purposes of this testing. The script measures the CPU usage for each of the processes and publishes and xml file that can be sent to the hypervisor of the cloud.

Conclusion

In order to evaluate the cloud computing providers based on the availability a new model was proposed in the Thesis. The model uses a new, innovative cloud architecture and software that enables the users to provision all of its instances, no matter which providers they are using at the same place. The model is based on perpetual communication between a virtual machine of a cloud and a central system in charge of provisioning the virtual machines.

The second part of this work is an experimental analysis of the causes of a failure of virtual machines. A test, known as stress testing, was performed on four machines of the leading cloud computing providers to determine which type of CPU load threatens to set threatens to cause a failure of a machine. The results show that the loads with an average growth rate that is less than 0.1% are usually not a threat to a machine, while those higher that 0.1% should be monitored at all times and, in some cases, terminated or moved to a more powerful machine.

These results however, can only be applied to the machines that were tested. The future work on this topic may include testing of a greater number of machines and obtaining figures that can be universally applied and used by various cloud computing providers. In addition, the availability was tracked for one month. The future work might include the same model of evaluation but with a longer period of availability measuring.

References

[1] NIST SP 500-292 NIST *Cloud Computing Reference Architecture*, http://www.nist.gov/customcf/get_pdf.cfm?pub_id=909505, September 2011

[2] GARTNER, *Gartner Says Smartphone Sales Grew 46.5 Percent in Second Quarter of 2013 and Exceeded Feature Phone Sales for First Time*, http://www.nist.gov/customcf/get_pdf.cfm?pub_id=909505, Egham, UK, August 14, 2013

[3] SIMON K., *Apps Solidify Leadership Six Years into the Mobile Revolution*, http://www.flurry.com/bid/109749/Apps-Solidify-Leadership-Six-Years-into-the-Mobile-Revolution#.U6_z5vl_tbp, Flurry, April 1, 2014

[4] GRAY, J. *Distributed Computing Economics*. Queue 6, 3 (2008), 63–68. Available from: http://portal.acm.org/ft_gateway.cfm?id=1394131&type=digital%20edition&coll=Portal&dl=GUIDE&CFID=19219697&CFTOKEN=50259492.

[5] AJAY O., Matlab-Mathematica-R and GPU Computing, *http://decisionstats.com/2010/09/21/matlab-mathematica-r-and-gpu-computing/*

[6] GARTNER, *Multitenancy*, http://www.gartner.com/it-glossary/multitenancy

[7] CARRARO, G., CHONG, F. *Software as a Service (SaaS): An Enterprise Perspective*, http://msdn.microsoft.com/en-us/library/aa905332.aspx, October 2006

[8] ANTONOPOULOS, N., GILLAM L., *Cloud Computing: Principles, Systems and Applications*, Springer, July 16, 2010

[9] ATHOW, D., *SaaS, PaaS and IaaS: Which Cloud Service Model Is For You?*, TechRadar Pro, January 14th 2014

[10] ARMBRUST, M.; FOX, A.; GRIFFITH, R.; JOSEPH, A. D.; KATZ, R. H.; KONWINSKI, A.; LEE, G.; PATTERSON, D. A.; RABKIN, A.; STOICA, I.; ZAHARIA, M.; *Above the Clouds: A Berkeley View of Cloud Computing*, Electrical Engineering and Computer Sciences, University of California at Berkeley, February 10, 2009

[11] MIJIC, M. et al., *Downtime Statistics of Current Cloud Solutions*, The Intersection of Cloud and Mobility, NIST, 25-27 March 2014

[12] LI, Z.; O'BRIEN, L.; ZHANG, H.; CAI, R., *On A Catalogue of Metrics for Evaluating Commercial Cloud Services*, Proc. 13th ACM/IEEE Int. Conf. Grid Computing (Grid 2012), IEEE Computer Society, Sept. 2012, pp. 164–173.

[13] AYERS, M. L., *Telecommunications System Reliability Engineering, Theory, and Practice*, Wiley-IEEE Press, December 2012

[14] AMAZON, *Amazon EC2 Service Level Agreement*, http://aws.amazon.com/ec2/sla/

[15] Akmai Technologies, *New Study Reveals the Impact of Travel Site Performance on Consumers*,

http://www.akamai.com/html/about/press/releases/2010/press_061410.html ,June 14, 2010

[16] HP Discover General Session Day 2 with Meg Whitman and Martin Fink, http://h30614.www3.hp.com/Discover/Events/LasVegas2014/SessionDetail/4ebe345b-de3e-4478-aa17-3878e67439b4

[17] ViFiB Company Profile, http://www.vifib.com/profile

[18] SlapOS Introduction, http://community.slapos.org/wiki/slapos-Wiki.Home/VIFIB-SlapOS.Introduction

[19] re6st Resilient Overlay Mesh Network, http://www.erp5.com/NXD-re6st.Two.Page

[20] Processor and CPU Time, http://www.gnu.org/software/libc/manual/html_node/Processor-And-CPU-Time.html#Processor-And-CPU-Time

[21] FERRARI, D. and ZHOU, S., *An Empirical Investigation of Load Indices For Load Balancing Applications*, Proc. Performance ’87, the 12th Int’l Symp. On Computer Performance Modeling, Measurement, and Evaluation, North Holland Publishers, Amsterdam. The Netherlands 1988. pp. 515-528

[22] Stress Testing, http://www.techopedia.com/definition/15310/stress-testing

Sažetak

Model predviđanja prekida rada kod računarstva u oblaku

Ovaj rad se sastoji od dva dijela. Prvi dio rada predstavlja novi model evaluacije pružatelja usluga računarstva u oblaku. Prvo poglavlje sadrži pregled terminologije računarstva u oblaku, drugo poglavlje predstavlja statistike dostupnosti nekih od najvećih pružatelja usluga. Treće poglavlje definira dostupnost pružatelja usluga računarstva u oblaku te zadnje poglavlje prvog dijela opisuje model stvarnovremenskog praćenja dostupnosti pružatelja usluga.

Drugi dio rada sadrži rezultate modela predviđanja prekida rada koji se temelji na eksperimentalnoj analizi iskorištenja procesora. Analiza uključuje stress-testiranje četiri virtualne mašine nekih od najvećih pružatelja usluga računarstva u oblaku.

Ključne riječi: računarstvo u oblaku, evaluacija, dostupnost, predviđanje prekida rada

Summary

Predictive Model of Failures in Cloud Computing

This work consists of two parts. The first part presents a new model of evaluation of the cloud computing providers based on the availability. The first chapter is a review of basic cloud terminology, the second presents the statistics of the current cloud solutions. The next chapter defines availability of the cloud computing provider and the last chapter of the first part describes the model of real-time availability tracking.

The second part of the Thesis shows the results of experimental failure prediction model based on the usage of Central Processing Unit. The testing includes stress testing four machines of some of the biggest cloud computing providers.

Keywords: cloud computing, evaluation, availability, failure prediction

Abbreviations

SaaS	*Software as a Service*
Paas	*Platform as a Service*
Iaas	*Infrastructure as a Service*
CRM	*Customer Relationship Management*
IWGCR	*International Working Group on Cloud Computing Resiliency*
RSS	*Rich Site Summary*
SLA	Service-Level Agreements
EC2	*Elastic Compute Cloud*
SlapOS	*Simple Language for Accounting and Provisioning*
CTO	*Chief Technology Officer*
HP	*Hewlett-Packard*
P2P	*peer-to-peer*
NIST	*National Institute for Standards and Technology*
OS	*Operating System*
CPU	*Central Processing Unit*
I/O	*input/output*

Appendix A: Detaljni opis stress-testiranja

Na svim virtualnim računalima je bila instalirana neka inačica operacijskog sustava Linux, kao i SlapOS. Za spremanje stanja sustava se koristila baza podataka tipa SQLite kako bi se odredilo koji procesi najviše oterećuju processor i RAM. Kako bi se izmjerilo prosječno opterećenje procesora, korištena je Python skripta kreirana u sklopu ovog rada. Skripta se pokreće naredbom python i trima argumentima: ime datoteke (report-tool.py), intervalom u sekundama (koji služi da bi se odredilo koliko često će system spremati stanje procesora) i imenom datoteke u koju se spremaju rezultati. Mjerenje opterećenja procesora se događalo u isto vrijeme kao i kad je izvršavanja kod alata "stress" operacijskog sustava Linux. Testiranje je običnot trajalo dok prosječno opterećenje ne bi premašilo broj procesora, a rezultati su spremani u dokument tipa .csv koji je kasnije kopiran na lokalno računalo i pretvoren u document tipa Excel koji je korišten za generiranje grafova i izračun prosječne stope rasta opterećenja.

Appendix B: Detailed Description of the Stress Testing

The machines that were used for the testing had some version of Linux operating system. The machines also had SlapOS installed. An SQLite database was used to store the snapshots of the system taken every minute in order to determine which of the processes/users were using the highest amount of RAM and CPU. For the purposed of the measurement of the CPU average load a Python script (created for this Thesis) was run at the same time the disk I/O was loaded with a Linux command “stress”. The script is run by the command python and three arguments: the name of the file (report-tool.py), an interval in seconds (that determines how often the system will take a snapshot of the CPU state) and the name of the file in which the results will be stored. The testing was usually run until the load surpassed the number of CPU’s and the results were stored in a .csv document that was later on copied to a local computer and transformed into an Excel document that was afterwards used to create graphs and calculated the average growth rates.