Criteria for the evaluation of private cloud computing

by

Piet Theron

Thesis presented in partial fulfilment of the requirements for the degree of Master of Science in Computer Science in the Faculty of Science at Stellenbosch University

Department of Mathematical Sciences,
Computer Science Division
University of Stellenbosch,
Private Bag X1, Matieland 7602, South Africa.

Supervisor: Prof. W. Visser

December 2013

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: 27 November 2013 ....

Abstract

Criteria for the evaluation of private cloud computing

P.J. Theron

Department of Mathematical Sciences,
Computer Science Division
University of Stellenbosch,
Private Bag X1, Matieland 7602, South Africa.

Thesis: MSc (Computer Science)

December 2013

Cloud computing is seen as one of top 10 disruptive changes in IT for the next decade by leading research analysts [1][2]. Consequently, enterprises are starting to investigate the effect it will have on the strategic direction of their businesses and technology stacks. Because of the disruptive nature of the paradigm shift introduced by it, as well as the strategic impact thereof, it is necessary that a structured approach with regard to risk, value and operational cost is followed with the decision on its relevance, as well as the selection of a platform if needed.

The purpose of this thesis is to provide a reference model and its associating framework that can be used to evaluate private cloud management platforms, as well as the technologies associated with it.

Uittreksel

Afrikaans

('Criteria for the evaluation of private cloud computing')

P.J. Theron

Departement Wiskundige Wetenskappe,
Afdeling Rekenaarwetenskap,
Universiteit van Stellenbosch,
Privaatsak X1, Matieland 7602, Suid Afrika.

Tesis: MSc (Rekenaarwetenskap)

Desember 2013

Wolk berekening word deur vooraanstaande navorsing ontleders [1][2] as een van die top 10 ontwrigtende veranderings vir IT in die volgende dekade beskou [1][2]. Gevolglik begin korporatiewe ondernemings met ondersoeke om te bepaal wat die invloed daarvan op hulle strategiese rigting en tegnologië gaan wees. Die ontwrigtende aard van die paradigma skuif, asook die strategiese impak daarvan, noodsaak 'n gestruktureerde ondersoek na die toepaslikheid en keuse van 'n platform, indien nodig, met betrekking tot risiko, waarde en operasionele koste.

Die doel van hierdie tesis is om 'n verwysings model, en 'n raamwerk wat dit implementeer, saam te stel wat dan gebruik kan word om privaat wolk berekening platforms te evalueer.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Prof. W Visser for his time spent and guidance throughout the project, and thank my family, in particular my wife Anita, and friends for their infinite support and patience throughout the time it took to finish the thesis.

Contents

List of Figures

List of Tables

Nomenclature

Acronyms

AHP Analytic hierarchy process [8]

AWS Amazon Web Services [9]

bps bits per second

Bps bytes per second

CMP Cloud Management Platform

CPU Central Processing Unit

DNS Domain Name Services

ECC Eucalyptus Cloud Controller [10]

IaaS Infrastructure-as-a-Service

ICMP Internet Control Message Protocol

IP Internet Protocol

KVM Kernel-based Virtual Machine [11]

MaaS Metal-as-a-Service

PaaS Platform-as-a-Service

PBR Performance Based Ranking

PCEF Private Cloud Evaluation Framework (This work)

PCMP Private Cloud Management Platform

RAM Random Access Memory

SaaS Software-as-a-Service

VM Virtual Machine

VMI Virtual Machine Image

Symbols

mean μ

StandardDeviation σ

Glossary

CSMIC The Cloud Services Measurement Initiative Consortium (CSMIC) is an initiative launched by the Carnegie Mellon University[12]. It is a consortium (Carnegie Mellon University, University of Melbourne, CA Technologies, KPMG, and others) of thought-leaders from educational institutions, end user organizations and technology providers, who are experts in measuring and managing IT-enabled services.

Framework The Cambridge Dictionary defines a framework as:

> *Framework is a system of rules, ideas or beliefs that is used to plan or decide something or a supporting structure around which something can be built [13].*

Hadoop A Apache open source project [14] delivering a framework for distributed computing. The core components consist of the Hadoop Distributed File System (HDFS) and the MapReduce [14] distributed computing engine.

Hypervisor A hardware virtualization technique allowing multiple guest operating systems (Virtual Machines) to run concurrently on a single host.

KVM Kernel-based Virtual Machine [11] is an open source hypervisor for Linux running on x86 hardware containing virtualization extensions (Intel VT or AMD-V).

LikertScale The Likert Scale is a rating scale used that allow individuals to express how much they agree or disagree with a particular statement. The scale uses Likert Items which is statements that the respondent is asked to evaluate. The following range will be used to rate KPI's in this paper:

1. Strongly disagree.
2. Disagree.
3. Neither agree nor disagree.

4. Agree.
5. Strongly agree.

load Average CPU load = demand for CPU.

OCCI Open Cloud Computing Interface [15] is an emerging standard defining Infrastructure-as-a-Service APIs which is delivered through the Open Grid Forum.

R An interactive environment and language for data analysis [16].

Referencemodel Wikipedia defines a Reference model as:

A reference model in systems, enterprise, and software engineering is an abstract framework or domain-specific ontology consisting of an interlinked set of clearly defined concepts produced by an expert or body of experts in order to encourage clear communication. A reference model can represent the component parts of any consistent idea, from business functions to system components, as long as it represents a complete set. This frame of reference can then be used to communicate ideas clearly among members of the same community. Reference models are often illustrated as a set of concepts with some indication of the relationships between the concepts.

UbuntuCloudInfrastructure Ubuntu cloud infrastructure [17] Provides IaaS based on OpenStack.

VM A virtual machine is an isolated guest operating system hosted in a Hypervisor.

Chapter 1

Problem statement

1.1 Introduction

Many IT organizations and IT divisions in enterprises find themselves with tighter budgets whilst they need to address an increased demand for agility and elasticity, in both hardware and software offerings. The traditional IT delivery models do not cater for these modern demands and cloud computing is surfacing as a suitable alternative to it [18][19].

Private cloud computing is of special interest to enterprises because of the trust it offers via on-premises control and ownership, as well as other cloud features such as scalability, automation, standardization, self-service and self-healing, to name but a few.

There is however a significant strategic impact and risk associated with a move from a traditional IT model to a cloud based model because of the vast differences between the models. With the traditional model the system administrators have full control over the environment and architectures, and technologies are mature with well-known usage and design patterns. With cloud computing the end user is empowered to manage components of the environment which can cause system administration issues. Architectures are different requiring a new way of thinking when solutions are designed and there are new technologies, big data and map-reduce programming for example, that software houses will have to conquer. Other factors that needs consideration is: management tools and processes; financial policies, models, and approaches; physical and virtual architecture design; security; storage; network engineering; ITIL and ITSM; and support services [20].

It is therefore essential that a structured approach with regard to risk, value and operational cost is followed with the decision on the relevance of cloud computing, as well as the selection of a platform if needed. In this paper we will investigate options for the definition of a framework that can be used for such evaluations.

1.2 Thesis layout

Chapter 1 (*this chapter*) gives an introduction to the thesis with an explanation on the purpose of the research and a brief overview of cloud computing in general, followed by a more in depth discussion on private cloud computing.

Chapter 2 (*Research design and methodology*) focuses on the research method followed, identifies evaluation criteria we feel necessary for the evaluation of Private Cloud Management Platforms (PCMP's), and give details on the test suite.

The test suite consist of use-case requirements, a test plan with detailed instructions, goals for the tests, tools to be used for test execution and instructions on how to rank the platforms for specific tests.

Details on the experimental set-up are also discussed as it represents a typical private cloud implementation.

The gist of the chapter is the definition of a reference model and our implementation thereof, the Private Cloud Evaluation Framework (PCEF), that can be used to score and rank PCMP's based on the viewpoints of management, system administrators and end users.

An easy to use technique to calculate quantitative scores is shown, and the chapter is closed with a step-by-step example on how to populate the PCEF.

Chapter 3 (*Systems Evaluated*) explains our choice of Hypervisor, Kernel Virtual Machine (KVM), and the rationale for the selection of the PCMP's that were chosen (OpenNebula [21], Eucalyptus [10] and OpenStack [22]) to test the PCEF with.

It also classifies PCMP's based on their features and explains where our test platforms fit into the classification.

This is the last of the research related chapters and the next section of the thesis will give a practical example on how to use the PCEF with an evaluation of the above platforms.

Chapters 4 and 5 (*Results: Systems administrator perspective* and *Results: End user perspective*) focuses on the performance tests from a system administrator and end user point of view.

They give details on the system administration capabilities of the PCMP's based on observations made during the evaluations. The bulk of the chapters covers the load tests performed and present details and an analysis of the results.

Chapter 6 (*Results Analysis*) investigates whether it is necessary to use the complete set of KPI's identified for the PCEF in Appendix A for an evaluation.

This angle needs investigation as the full set might be too comprehensive for all evaluations. There is a possibility that the quantitative performance related results, obtained during the load tests, might be sufficient for an evaluation.

Chapter 7 (*Related Work*) give summaries of existing efforts on the ranking of PCMP's as well as other cloud technology research efforts that assisted us in the shaping of our work.

Note: The majority of the references in this paper refer to web sites as cloud computing has not been fully adopted by large enterprises and academic institutions. It is currently driven by internet communities that are typically financed by Google, IBM and other prominent players in the IT community. There are thus only a few academic papers available on the topic with the majority of them focusing on the benchmarking of public clouds. These efforts are unsuitable for private clouds as it, rightfully so, ignore the system administration component of cloud platforms.

Chapter 8 (*Conclusions and future work*) concludes the paper and suggest future work based.

Appendices

Appendix A give details on the complete set of Categories, Attributes and KPI's that can be used in the PCEF.

1.3 Objectives of thesis

The following must to be in place to successfully evaluate PCMP's:

- A guideline on the characteristics of private cloud computing so that there is no uncertainty on the term '*PCMP*'.
- A reference model for the evaluation of PCMP's based on the requirements of different viewpoints.
- An implementation of the reference model that must include, but not be limited to:
  - A test suite that includes tools and a methodology that can be used to test various aspects of PCMP's.
  - A unique set of KPI's that was shown, during the tests, to identify weaknesses and differences between PCMP's.
  - A scoring mechanism that clearly indicates whether a system is suitable for a specific use-case as well as which system is preferred.

In this thesis we will attempt to resolve all of the above with an in-depth investigation into private cloud computing.

1.3.1 Scope

We will cover all the major cloud service models, Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS) but only a subset, private clouds, of the cloud deployment models. Public clouds and hybrid clouds, a combination of private and public clouds, are out of scope for this thesis.

Comparative tests against Amazon EC2 [23] will however be performed to compare the performance of VM's deployed on the PCMP's we evaluated, to that of similar VM's deployed on Amazon EC2. This information might be useful to link our work to other studies.

1.4 Definition of cloud computing

Cloud computing is a concept that promise flexible and scalable services anywhere/anytime, embracing automation, on-demand self-service, elasticity and commodity hardware. It essentially boils down to a way of provisioning IT resources (e.g., networks, servers, storage and applications) from dynamic pools of virtualized resources.

In an attempt to standardize cloud computing, the National Institute for Standards and Technology [3] has come up with the taxonomy visualized in *Figure 1.1*

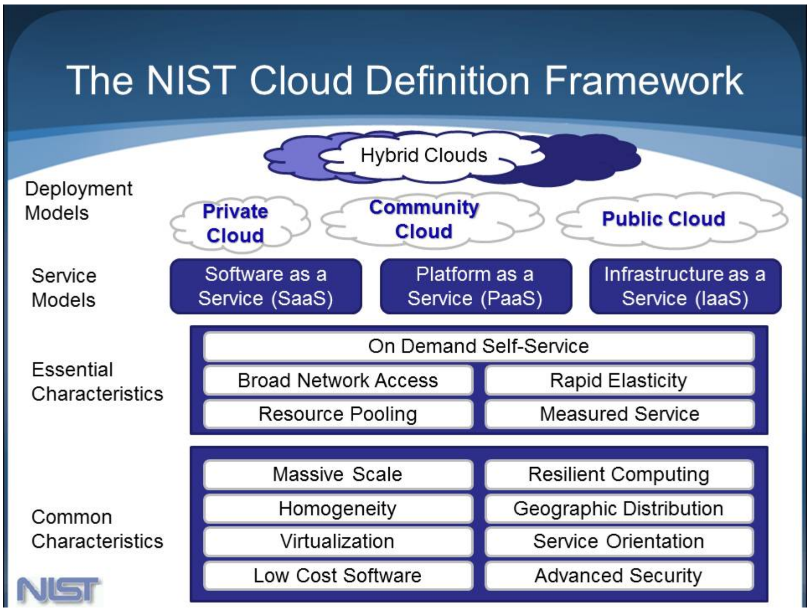


Figure 1.1: NIST cloud Framework [3]

Deployment models: The three major categories of cloud deployment models are:

- **Private clouds** which are the safest model for enterprises as all of the infrastructure and services are hosted in the perimeter of the enterprise. It is also a natural evolution from virtualization efforts already in place and offers the strongest security and audit capabilities of all of the cloud deployment models. It does however, when compared to public clouds, come with a larger up-front investment and is unlikely to have the same scalability and elasticity than that of public clouds.

- **Public clouds** which are a shared infrastructure model where all infrastructure and services are hosted by an external service provider, resulting in a low up-front investment, greater cost savings and hands-off administration, but it introduces legislation and audit challenges as data can be hosted in a number of physical locations.

- **Hybrid clouds** which is a combination of private and public clouds.

Services models: The three major categories of cloud service models are:

- **Software-as-a-Service (SaaS)** is publicly accessible software deployed on a remote system. In this category the cloud user typically manages the use of an application only, with the rest of the infrastructure managed by the service provider.

  Google Mail is a well-known example of a SaaS offering.

- **Platform-as-a-Service (PaaS)** is one level down from SaaS and provides the end user with the hardware infrastructure, Virtual Machines (VM's) including the operating system and middle-ware to be used, and components needed to deliver the end user's product or service.

  In the PaaS model, the end user uses tools and libraries from the service provider to build the software which will be deployed on the PaaS platform. The end user is responsible for the management of the application and its associated data, and the service provider is responsible for all other administration aspects of the system.

  Google AppEngine and Microsoft Azure are examples of PaaS offerings.

- **Infrastructure-as-a-Service (IaaS)** is one level down from PaaS and essentially provides end users with raw computing resources; the hardware infrastructure and the virtualization layer (hypervisor) end users will need to run their application on. The end user is responsible for the management of the VM and all software, including the operating system and middleware, needed by their applications.

Amazon EC2, Red Hat OpensStack, Eucalyptus and OpenNebula are examples of IaaS offerings.

Figure 1.2 summarizes the above with an overview of the roles and responsibilities for the cloud service models.

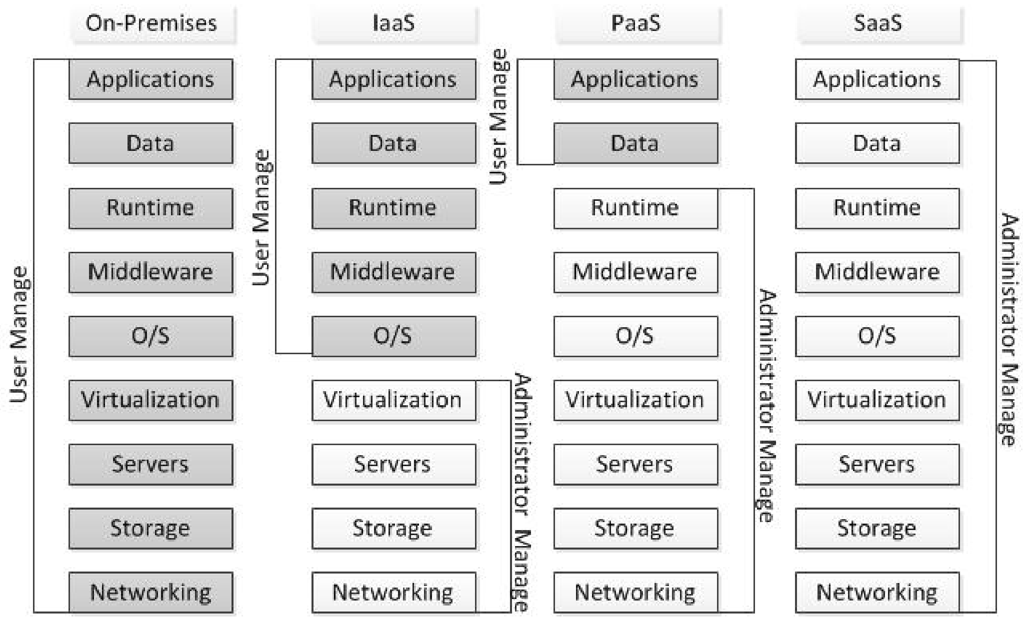


Figure 1.2: Cloud service model comparison grid [4]

1.5 Definition of private clouds

For a platform to qualify as a PCMP it must provide us with virtualization, broad network access, automation and on-demand self-service capabilities.

Virtualization

Virtualization is the foundation for cloud computing because of its resource pooling and rapid elasticity capabilities, but its benefits are limited if used on its own. It needs supporting systems for it to be fully effective in the cloud environment. These supporting systems should include, but not be limited to:

- An inventory system that keeps track of VM's and licenses issued as well as the resource allocations, such as disk, memory and CPU's. The system should also provide trigger points that can be attached to monitoring and diagnostic systems to assist in capacity management.
- A mechanism that enables self-healing so that maintenance and failures are transparent to users. For example, the ultimate outcome for an end user in the event of any system failure is zero downtime and no interruption in service.
- An operational management system to monitor and diagnose the data center. This is essential for the day-to-day operational requirements of any data center and should be provided out of the box.

Broad network access

Cloud computing is managed on-line from both a system administrator and end user's point of view. The cloud technologies themselves are designed around distributed computing models and a stable network with ample bandwidth is thus required for the operation of a private cloud.

Automation

Automation of human tasks via a workflow system is essential to a cloud environment as it is usually the system administration tasks that are the bottleneck in a virtualized only environment. Examples of such tasks are: Virtual Machine (VM) management, software and platform installs, patch management, security, network and storage configuration and firewall configuration.

On-demand self-service

Self-service is in our opinion the most important aspect of private clouds and it should offer:

- A self-service portal that allows users to manage cloud resources.
- Standardized service offerings such as pre-configured VM's for standardization purposes.

- A consumption tracking and billing system (metering) that can track and report on resource utilization. This is important because current virtualization technologies introduce ease of server commissioning that usually results in orphan virtual servers that is not needed anymore. These servers unnecessary consume resources such as disk space, CPU, memory, software licenses, etc. This phenomenon is known as VM sprawl [24] and it is clear that VM's are becoming the *"tmp"* files of IT systems; a feature that needs firm management.

1.5.1 Private cloud models

Llorente [5] categorize private clouds, based on their features, as follows:

- Data center virtualization which is an extension of virtualization; VMWare's vCloud [25], for example, is an infrastructure automation tool to orchestrate and simplify the management of VM's that fits into this category.

- Infrastructure provisioning which is similar to Amazon's EC2 model [23] for the provisioning of on-demand virtualized resources.

Private clouds will typically be built on the IaaS service model which means the platform used should be balanced between these two models as it would have to support both the system administrators and end users, see Figure 1.2.

The system administrators will prefer the data center virtualization model as it will assist them with the management of the VM's, whereas the end users will prefer the infrastructure provisioning model as it is geared towards the management of the configuration, life cycle and management of services.

Section 3.3 gives a more detailed explanation of the philosophies.

1.6 Summary

In this chapter we justified the research effort, gave a high level overview of the thesis, identified the expected outcome of the research, gave an overview of cloud computing and explored the details around private cloud computing.

Chapter 2

Research design and methodology

2.1 Introduction

This chapter focuses on the research method followed, the evaluation criteria needed to evaluate PCMP's, the test suite, the experimental set-up, and then the proposed reference model and framework for the evaluation of PCMP's.

It also give brief overviews of the related work used as the foundation for our work, and the mechanics of the scoring system developed for the PCEF.

The final sections give details of a set of critical KPI's identified during the load tests and closes off with a step-by-step example on how to populate the PCEF.

2.2 Related work and PCEF overview

The original intent of our work was to create an evaluation framework that consists of a generic set of evaluation criteria and metrics, an evaluation methodology, tools that can be used for performance related tests, and to use the Likert scale across all the metrics as a scoring mechanism.

During the related work research, however, we came across the work of the Cloud Services Measurement Initiative Consortium (CSMIC) [26] via the work done by Garg *et al.* [6] in which they propose a framework, the SMICloud, for the comparison and ranking of cloud services.

CSMIC

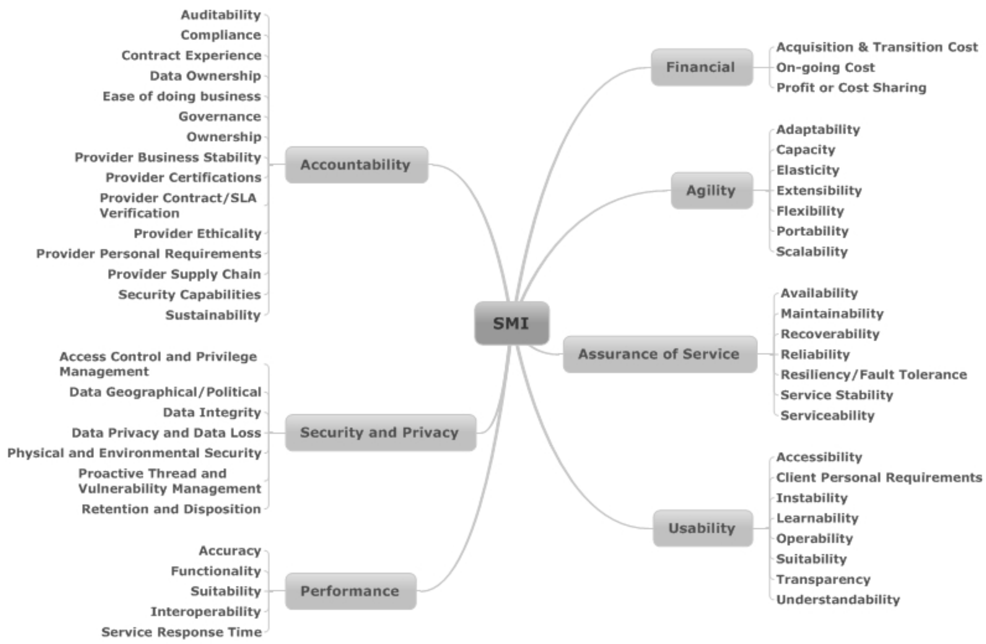
The CSMIC is busy with a standardization effort in which they define criteria that can be used to evaluate CMPs. They define a hierarchical framework, the Service Measurement Index (SMI), that consists of Categories, Attributes and KPI's, or measurements, that can be used to score platforms. At the time of this proposal the SMI was a work-in-progress with the Categories and Attributes defined, but the KPI's still outstanding.

SMI categories:

	Category	Questions
1	Financial	How much does the PCMP cost?
2	Accountability	Can we count on the software provider, organization and community?
3	Agility	Can the PCMP be changed and how quickly can it be changed?
4	Assurance of Service	How likely is it that the PCMP will work as expected?
5	Performance	Does the PCMP do what we need?
6	Security and Privacy	Is the PCMP safe and is privacy protected?
7	Usability	Is the PCMP easy to learn and to use?

Table 2.1: The 7 top-level Categories of the CSMIC SMI [7]

SMI attributes:



Categories with Attributes [7]

SMICloud

The SMICloud defines a methodology, based on the Analytic hierarchy process (AHP) [8], to score cloud services with a combination of the KPI scores and priorities assigned to the elements in the SMI KPI, Attribute and Category layers. These layers contribute to the AHP hierarchy used in the scoring.

Relationship to the PCEF

Our work build on the above efforts, see Section 2.5.2 for a detailed explanation, by extending the SMI with the addition of a top level layer that classify the final score according to the requirements of management, system administration staff and end users in order to acknowledge that there are different viewpoints involved with the selection of a PCMP.

We also propose a set of KPI's, that can be used with the SMI, which consist of the result of our initial work on the evaluation criteria, as well as other KPI's identified during the platform evaluations, as it seems to be a natural fit for the SMI KPI's.

In addition to the extensions of the SMI, the PCEF also deviate from the SMICloud with the adoption of a different KPI scoring mechanism. The SMICloud KPI scoring mechanism is elegant from a mathematical point of view, but it seems to be an overkill for the problem at hand.

Our KPI scoring mechanism is an easy to use technique that fairly compares KPI's with different measurement units to each other. The KPI scores are normalized to a percentage value making it easy to interpret. It was evaluated against the SMICloud's scoring mechanism, by using the same data used in their paper [6], and the PCEF achieved a similar result as them.

The final scores for platforms evaluated with the PCEF will also use the AHP with different weights (priorities) assigned to each of the attributes, categories and viewpoints, a KPI belong to.

PCEF illustration

Figure 2.1 gives a graphical representation of the PCEF AHP hierarchy.

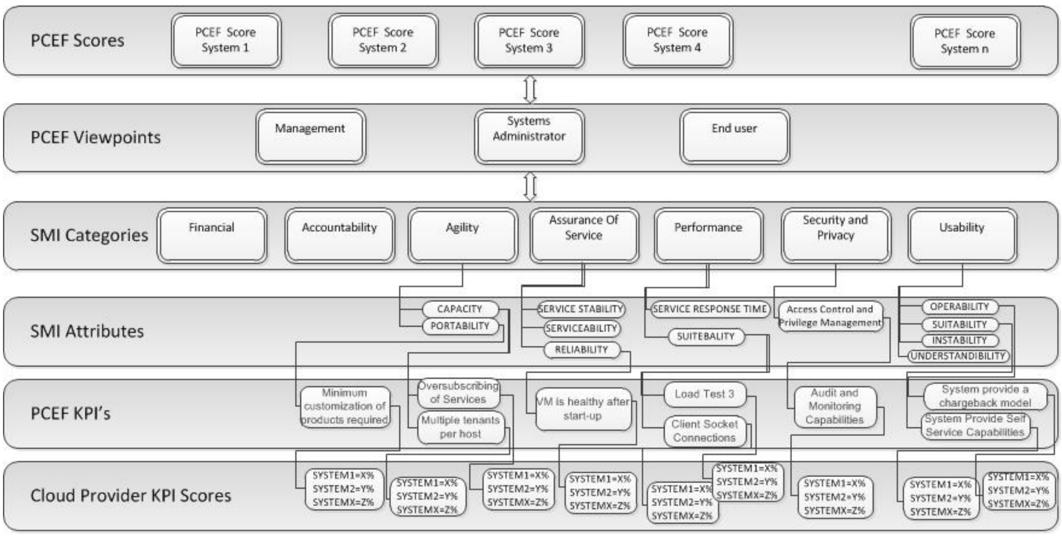


Figure 2.1: Private Cloud Evaluation Framework AHP hierarchy

Populating the PCEF is done by first calculating a normalized score for a platform on a KPI level. This score is then multiplied with the 'PCEF KPI' priority, then multiplied with the 'SMI Attribute' priority, then multiplied with the 'SMI Category' priority, and finally multiplied with the 'PCEF viewpoint' Priority, see Section 2.6.1 for an example.

The final value is the KPI's contribution to the score for a platform. The sum all of the values give the final score for a platform.

The PCEF is flexible enough to use with any number of the above top level views or with any subset of the lower layers, for example, the scoring obtained from the tests in the test suite, see Section 2.5.3, can be used as the only KPI's if you are interested in performance results only.

2.3 Research method

Our research method presented in *Figure 2.2* entailed the following steps:

1. Definition of the reference model.

2. Selection of the KPI's to use for the scoring in the PCEF.

   The selection of the KPI's involved extensive research into which criteria to use when infrastructure and IT vendors are evaluated.

3. Selection of diagnostic and monitoring tools.

   A lot of effort went into the selection of the monitoring and diagnostic tools as it is crucial for an experimental system like this to understand, at all levels (operating system, applications, network, etc.), what the state of the system at any given point of time is.

4. Definition of a test suite.

   This step involved research on system administration for hardware infrastructure, virtualization and PCMP's. The research included topics such as security, resource utilization to mention but a few.

   The combined result of this, and the previous phase was a comprehensive testing and benchmarking platform that included a database that was used to capture the results during the experiments, a tool-set that covered resource monitoring as well as the realization of the testing framework.

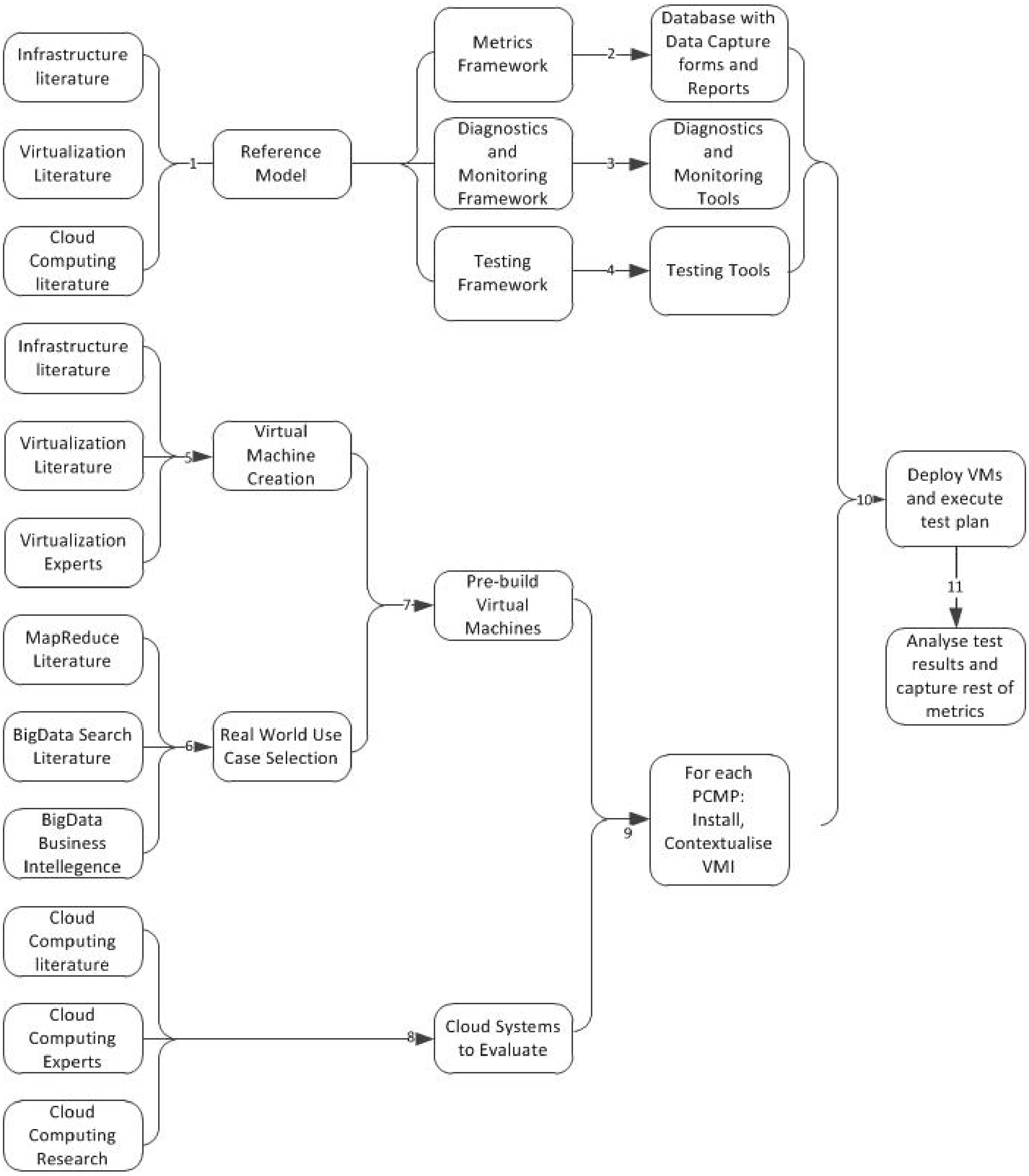


Figure 2.2: Research method

5. Virtual machine image (VMI) creation.

   The next topic to master was the creation of VMI's that would be compatible with PCMP's deployed on x86 infrastructures. We chose x86 because of its strong presence in enterprizes, its low cost if compared to that of mainframes, as well as the availability of PCMP's for x86 architectures.

Cloud on microprocessors, ARM for example, were also investigated but we decided to proceed with cloud on x86 as it have a better change to be adopted in enterprises at present. This might change in future because of the current interest in micro servers by hardware manufacturers [27][28][29]. These systems are starting to surface however [30] and TryStack [31] offers a free sandbox environment, for testing purposes only, that supports OpenStack on ARM processers.

The initial intend of this phase was to get an in-depth understanding of virtualization, but it ended up as a pre-requisite for the experiments because of the lack of VMI creation facilities with our PCMP's. Various VMI creation mechanisms were tested and the Linux command line tool kvm was selected because it proofed to be the most stable.

The end result was a recipe for Kernel-based Virtual Machine (KVM) VMI creation that was used to create a VMI that could be used on all the platforms tested.

6. Selection of real world use cases.

   We acknowledge the fact that the load tests are boundary tests representing worst case scenarios and therefore include real-world use cases for a more realistic view on the platforms as well.

   Two use-cases were selected to test and demonstrate the real-world capabilities of the PCMP's.

   The first use-case involves the statistical analysis of log files captured by an Enterprise Service Bus (ESB) for trend analysis and Service Level Agreement (SLA) management purposes. RHIPE [32], a merger of R [16][33] and Hadoop [14][34], was chosen for this use-case. It has a steep learning curve and mastering it consumed a large portion of the research phase.

   This test will represent a distributed system focusing mainly on CPU and memory resources. R, the processing component, is an in-memory application.

   The second use-case selected was that of a Web server. The Apache [35] and Apache Tomcat [36] open source web servers were chosen for this use-case. The pages served was a combination of static and dynamic pages resulting in a good spread of processing power needed across the processor, memory, disk and network. Apache benchmark [35][37] was used to emulate the web clients.

7. Creation of a pre-built VMI.

   The recipe from step 5 was used to create a generic VMI that was used on each of the PCMP's tested. The VMI contained the operating system, all the diagnostics and monitoring tool's clients, as well as the software used in the test suite and real world use-cases. It needed only minor changes in one of the startup scripts, for network configuration purposes, before deployment on the platforms.

8. PCMP Selection.

   The next step was to select the PCMP's that could be used to evaluate the effectiveness of the PCEF. Numerous cloud computing resources [38][39][17][22][21][and more] were consulted for this phase.

9. PCMP Installation.

   The ground work phase, discussed above, laid the foundation for the experimental phase which involved the installation of the platforms, contextualizing of the VMI for usage with the platform in use, and to learn how to administer and use the platforms.

10. Test execution.

    The experiments entailed the execution of administration, load and real world tests as described in in Section 2.5.3.7.

11. Results analysis.

2.4 Reference model

The reference model has two main components:

- A taxonomy which defines terminology, and provides a coherent description of the components and conceptual structure of the evaluation framework.

- Associated graphics, which provides a visual representation of the taxonomy, as an aid to understanding.

The model will be realized with an assessing framework, the PCEF.

2.4.1 High-level overview

The primary requirement for the assessing framework is that it must be able to evaluate PCMP's based on the requirements of different viewpoints. The end result of the evaluation must be a report that represents the suitability and performance of the platforms evaluated.

The reference model, Figure 2.3, is based on standard concepts associated with the evaluation of infrastructure and end user systems and has four fundamental elements:

- The definition of stakeholder viewpoint specifications.
- The definition of an infrastructure evaluation and monitoring specification.
- The definition of the evaluation criteria specifications used to score the platforms with.
- A scoring methodology that can be used to assess PCMP's.

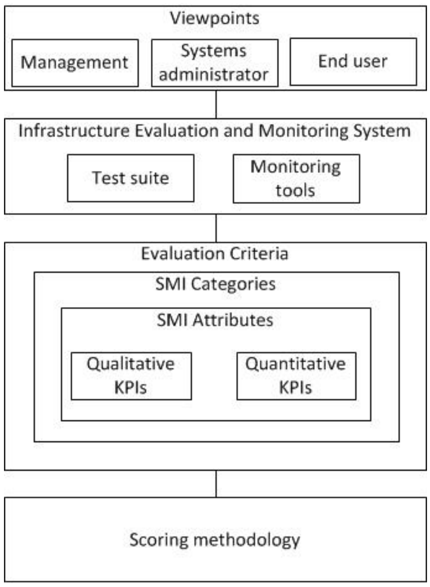


Figure 2.3: Reference model

2.4.2 Viewpoint specifications

A viewpoint, in the context of our work, is a subset of the criteria that brings together the relevant information needed by a stakeholder to make an informed decision.

2.4.3 Infrastructure evaluation and monitoring specification

Efficient testing and monitoring of the infrastructure is crucial to the success of the framework. A system that can monitor and record pre-defined system metrics during the tests is therefore required. The tests should be provided by a test suite and the test cases should satisfy test coverage across all the relevant criteria identified by the evaluation criteria specification. The tests must also have the ability to be automated or it should be possible to easily modify it to be automated. This strategy allows for quicker and repeatable testing while reducing the risk of human error.

2.4.4 Evaluation criteria specifications

The Evaluation Criteria must be comprehensive enough to satisfy the needs of all the viewpoints and should include both quantitative and qualitative measures.

2.4.5 Scoring methodology

The scoring methodology must be able to fairly compare metrics with different measurement units. This is necessary because low level value metrics, such as CPU load, in combination with high level value metrics, like the disk I/O throughput, will be used to calculate a score for a platform.

2.5 Private Cloud Evaluation Framework

The sections following below describe the implementation framework of the reference model above. Each component will be discussed individually because of the level of detail involved.

2.5.1 Viewpoint specification

The PCEF cater for three viewpoints:

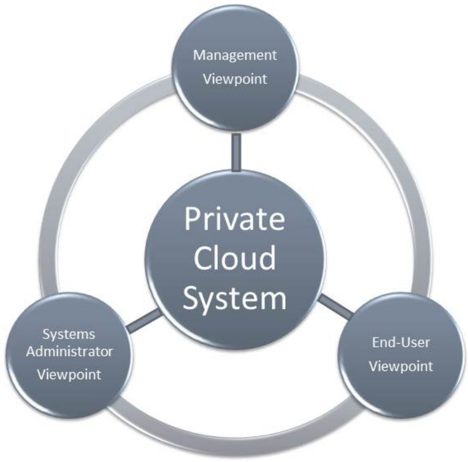


Figure 2.4: Viewpoint model for the PCEF

- The management viewpoint, which focuses on the purpose, financial requirements, accountability and security of the system. It represents the business requirements and defines the KPI's needed to define and measure them.
- The system administrator viewpoint, which focuses on the performance, administration, agility and accountability of the system. It represents the system administration requirements and defines the KPI's needed to define and measure them.
- The end users viewpoint, which focuses on the usability, security and privacy, performance and service assurance of the system. It represents the cloud platform's end users requirements and defines the KPI's needed to define and measure them.

2.5.2 Scoring: Percentage Based Ranking (PBR) mechanism for the results

A simple technique to fairly compare KPI's with different measurement units is to normalize the results to a percentage based value. This is achieved by calculating the percentage difference between the measured value and the best value achievable/required for a test.

The initial implementation of the mechanism used the best test result obtained across the platforms as the baseline value, but this technique does not cater for a suitability assessment. For example, if all the platforms being evaluated get the same, low score on a critical KPI, then this failure will be hidden because the normalization process will score the platforms against each other, each ending up with the same score, 100%.

The scoring mechanism was therefore altered to use an 'Expected Score', with the same scale as the KPI, instead.

The expected value can for example be the best result between the platforms tested, a pre-determined theoretical value, a baseline test result or the maximum value if the Likert scale is used.

The formulas below describe the calculation.

Note: It should be noted that the formulas are designed for values ≥ 0.

1. Choose an expected result for a test.

   If the expected value is zero, then select a maximum value that would be acceptable

2. Normalize the test results:

   If a higher value is better, then

   $$\bar{x}_i = \frac{Observed_i}{Expected_i} * 100$$

   else

   $$\bar{x}_i = \frac{Expected_i}{Observed_i} * 100$$

   In the special case of a zero expected value the maximum allowed deviation from zero is used as the measure of relative performance.

   $$\bar{x}_i = 100 - \frac{Observed_i}{Maximum_i} * 100$$

   Where

   $\bar{x}_i$ - Normalized value

$Observed_i$ = Observed value
$Maximum_i$ = Maximum allowed deviation from zero for the test associated with $Observed_i$

3. Calculate the average across all scores to get the final score:

$$\mu = \frac{\sum_{i=1}^{N} \bar{x}_i}{N}$$

Where
μ = Final score
$\bar{x}_i$ = Normalized value
N = the number of tests

Note: Any reference to the ranking of a test result from here-on will refer to the percentage based ranking mechanism unless otherwise stated.

2.5.3 Test suite

The test suite compiled for this work is a comprehensive testing and benchmarking platform that contains:

- A collection of system administrator and end user performance and real-world use-case requirements that covers the usability of the PCMP's from both a hosting infrastructure administrator as well as a end users perspective.
- A test plan that includes detailed instructions to be followed to execute the tests as well, as the goals for each of the test cases.
- The tools that can be used to perform the tests as well as the tools needed to monitor the platforms during the tests.

2.5.3.1 Platform administration requirements

Administration of the PCMP's is an important aspect from both the system administrator and the end user's point of view. Both of them need an easy to use user interface to perform their daily activities effective and efficiently. The following paragraphs provide details on these duties.

System administrator requirements The objective of the system administrator assessment was to determine whether the tools provided by the PCMP's were sufficient to satisfy the operational requirements needed by system administrators.

The criteria used were based on the following:

- System administration capabilities including installation of the system, day-to-day management tasks, capacity planning and diagnostics capabilities.
- VM management capabilities.
- Platform performance when stressed.
- Capability for over-subscription of CPU and memory resources.
- Reporting capabilities.
- Cloud controller overhead and behavior.

End user requirements The objective of the end user administration assessment was to determine whether the tools provided by the PCMP's were sufficient to satisfy the operational requirements needed by end users.

The criteria used were based on the following:

- Self-service capabilities including the user interface that must be used.
- VM management capabilities.
- VM monitoring capabilities.

2.5.3.2 Load and functionality testing tools and coverage

The following tools and test suites were selected to evaluate the aspects of PCMP's.

The 'Linux Test Project (LTP)' [40] An automated Linux kernel test suite that will be used to firstly validate the reliability, robustness and stability of the VMIs' operating system when deployed on the PCMP, and secondly, to get some performance statistics. The LTP was not designed to be a performance benchmark but it does stress the kernel sufficiently during its tests for us to collect useful data.

The execution time for the default LTP is too long to be practical and the test was trimmed down to only execute a subset of the tests. The following tests were selected:

- Math library tests (CPU tests) that calculate absolute values, convert ASCII to floating point values, solve Bessel's differential equations, and perform exponential/logarithmic, power, square root and trigonometry calculations.
- Memory management tests, using the mmap Unix system command (it maps files into memory), to create and destroy a 10000 page memory map, repetitively creating a 1 page map for one minute, and check for race conditions.
- Scheduler stress tests that create, run and destroy multiple threads.
- Native POSIX Thread Library (NPTL) tests that perform thread related tests using mutex locks.
- File system load tests that grow/shrink files as well as executing numerous read/write tests.

bonnie++ [41] A file-system benchmarking suite designed for hard disk and file system tests, was selected for its system resource utilization and performance testing capabilities.

The test suite is designed to contain real-world disk and file-system tests that benchmark the following:

- Sequential output (Per character, block and rewrite).
- Sequential input (Per character and block).
- Random seeks.
- Sequential create (Create, read and delete).
- Random create (Create, read and delete).

The first three categories, according to the documentation, simulate the same kind of I/O load a database would normally put onto a disk. The last two categories is meta-data tests that simulate the kind of load you would typically expect to see in a web caching server. For these tests files with a zero byte size are created, read, and finally deleted.

netperf [42] A networking benchmark designed to measure network performance for the following environments:

- TCP and UDP via BSD Sockets for both IPv4 and IPv6.
- DLPI.
- Unix Domain Sockets.
- SCTP for both IPv4 and IPv6.

Apache Benchmark (ab) [35][37] A tool for benchmarking Hypertext Transfer Protocol (HTTP) servers

Phoronix Test Suite [43] A comprehensive testing and benchmarking test suite based on the extensive range of testing and internal tools developed by Phoronix.com.

The following subset of the tests available was used to test the CPU and memory:

- memory.

  This memory benchmark consists of tests designed to test the RAM performance, including speed and bandwidth.

- compress-gzip.

  This memory and CPU benchmark measures the time needed to compress a 2GB file using Gzip compression

- compress-7zip.

  This memory and CPU benchmark uses the 7-zip compression algorithm and shows a rating in million instructions per second (MIPS)

- C-Ray.

  This CPU benchmark is a multi-threaded test designed to stress test the floating-point processor. The time for the test to complete will be used as the KPI value.

Focus areas for testing tools:

	Load and Stability	CPU	Memory	Disk	Network	Real-world
LTP	X	X	X	X		
Web servers	X	X			X	X
RHipe	X	X	X			X
netperf	X				X	
bonnie++	X			X		
Phoronix g-zip	X	X	X	X		
Phoronix 7-zip	X	X	X			
Phoronix C-Ray	X	X				
Phoronix memory	X		X			

2.5.3.3 Hosting- and VM testing

The objectives of the tests are to assess the behavior of the PCMP's when there is a load on the hosting nodes and/or deployed VM's from both the end user and system administrator's viewpoint. The assessment monitors the following:

- CPU load related behavior; for example the load a VM induces on the underlying infrastructure when in idle mode as well as when it is executing applications.
- Performance penalties introduced on a VM by the PCMP (cloud controller and virtualization layer) and co-hosted VM's.
- VM stability and performance under normal and stressed conditions.
- Overall stability of the system.

Notes:

- The maximum CPU load on a machine should ideally stay below the amount of CPU cores available. For example, if you have a single dual-core CPU than the CPU load should not exceed 2.
- The management viewpoint is not covered by the load tests as management only need confirmation on the suitability of new technologies when it is introduced.

Tools used: bonnie++, Phoronix, LTP and netperf

The following load tests were executed:

- Load test 1:

  Objective

  A bare-metal test on the hosting machines, with no VM's deployed, to capture a baseline result that will give us the best result achievable on the hardware used. The test results can be used as the expected values for the percentage based scoring if you want to measure the impact of the PCMP's overhead.

  Method

  Run bonnie++, Phoronix, LTP and netperf on all of the hosting machines and use the average results as the baseline value.

  Ranking

  Save the average of the results as a baseline for future rankings.

  Viewpoint

  Baseline

- Load test 2:

  Objective

  This test is similar to that of Load test 1 but there will be idle VM's deployed on each of the hosts. The objective of this test is two-fold:

  - To test the effect of idle VM's on a host by comparing the host's load figures to that of Load test 1.
  - To test the impact of a host under load on an idle VM by measuring the load on the VM. This would give an indication of the end user experience when the VM host is stressed by either standard system administration tasks (back-ups, etc.) or co-hosted VM activities.

  Method

  1. Deploy one VM with a large configuration on each of the hosts.
  2. Run bonnie++ on the host and record the load on the hosts and VM's. *Note:* Additionally to the load test, we also logged onto the VM and executed simple tasks to test the responsiveness of it.
  3. Deploy multiple VM's on a host and repeat step 2.

  Ranking

  - Host ranking: Rank the hosting machines by using the baseline result obtained in Load test 1 as the expected value.
  - VM ranking: Rank the VM's by using the best result achieved across the platforms as the target value.

  Viewpoint

  End user and system administrator

- Load test 3:

  Objective

  The primary objective of the test is to get a baseline result for a VM under favorable conditions by stressing it whilst there is no other load on the hosting machines. The secondary objective is to measure the impact of a stressed VM on a host by measuring the load induced by the VM on it. This info can be used by system administrators for capacity planning purposes.

  Method

  The tests were run on each of the VM's deployed in Load test 2 and the average of the results was saved as the baseline VM result.

1. Deploy one VM with a large configuration on each of the hosts.
2. Run bonnie++, LTP, and the Phoronix Test Suite on the VM.

Ranking

- Host ranking: Rank the hosting machines using the loads recorded on it (VM induced load) using the best result as the expected value.
- VM ranking: Rank the VM's by using the bare-metal results obtained in Load test 1 as the expected value. (to get an indication of the performance loss via the virtualization layer).
- VM ranking: Rank the VM's against each other across all tests using the best result as the expected value.

Viewpoint

End user and system administrator

- Load test 4:

Objective

For this test bonnie++ was ran simultaneously on the VM's and hosts. We focused only on bonnie++ as we foresee, based on experience in similar environments, that it will be primarily system backups that will be affecting the VM's. The primary objective is to test whether the cloud platforms can handle a stressed environment when both the hosts and deployed VM's are stressed. The secondary objective is to test the effect of a host and VM fighting each other for resources.

Method

1. Deploy one VM with a large configuration on each of the hosts.
2. Run bonnie++ simultaneously on the VM's and hosts.

Ranking

- Host ranking: Rank the platforms by scoring them against each other using the best results across the tests as the expected value.
- VM ranking: Rank the platforms based on the throughput for bonnie++ by using the best value obtained in Load test 3 as

the expected value and record the load on the hosts for the system administration viewpoint.

Viewpoint

End user and system administrator

- Load test 5:

  Objective

  For this test two VM's are deployed on a host; one in idle mode and on stressed . The primary objective is to test the effect of a stressed VM on co-hosted VM's.

  Method

  1. Deploy two VM's with a medium configuration on each of the hosts.
  2. Run bonnie++ on one of the VM's.

  Ranking

  VM ranking: Rank the platforms based on the load on the idle VM's as well as the bonnie++ throughput on the active VM using the best baseline result as the expected value and record the load on the hosts for the system administration viewpoint.

  Viewpoint

  End user

- Load test 6:

  Objective

  The primary objective was to test the effect of multiple stressed VM's on each other by executing bonnie++ simultaneously on the two VM's used in Load test 5.

  Method

  1. Deploy two VM's with a medium configuration on each of the hosts.
  2. Run bonnie++ simultaneously on both of the VM's.

  Ranking

- VM ranking: Rank the platforms based on the load on the VM's by using the best result as the expected value.
- VM ranking: Rank the platforms based on the bonnie++ throughput, using the result achieved in Load test 3 as the expected value.

Viewpoint

End user

2.5.3.4 Network testing

The next set of tests to perform is the network throughput testing.

Tools used: netperf

Test cases:

1. Network test 1:

   Objective

   A bare-metal test to measure the maximum throughput between hosts to get a baseline result set for maximum network throughput.

   Method

   Execute netperf: host-to-host

   Viewpoint

   Information purpose only

2. Network test 2:

   Objective

   Test the throughput between the hosts and local VM's to get a baseline result set for maximum VM throughput. This test will also measure the penalty paid for going via the virtualization layer.

   Method

   Execute netperf: host-to-local-VM

   Viewpoint

Information purpose only

3. Network test 3:

   Objective

   Test the throughput between the VM's

   Method

   Execute netperf: VM-to-VM

   Ranking

   VM ranking: Rank the platforms against each other by using the best result as the expected value.

   Viewpoint

   End user

2.5.3.5 Real-world use-case testing

The next set of tests performed was to test the real-world applications as described in the 'Research method' section, deployed on the PCMP's.

The objective of these tests were to get an understanding of the performance of the platforms based on real-world examples.

Tools used:

- Apache web server [35].
- Apache Tomcat web server [36].
- RHipe [32] (R [16], Apache Hadoop [14] and MapReduce [14]).

Test cases

1. Web server test:

   Objective

   Test the performance and impact of a web server deployed on a VM

   Method

Install the Apache and Tomcat Web Servers on the VM and test the servers with a load test tool, Apache Benchmark [35][37] in our case, from outside the cloud platform

Ranking

VM ranking: Rank the platforms by comparing the transaction rates as well as the number of successful client connections. The average of the 2 PBR scores is used as the final score.

Viewpoint

End user

2. Hadoop based statistical analysis application

   Objective

   Test the performance and impact of statistical analysis tool deployed on the cloud

   Method

   Install the Apache Hadoop and RHipe packages on the cloud, copy log files from outside the cloud to the cloud file-system and do a statistical analysis with RHipe on it.

   Ranking

   Rank the platforms by comparing the time it takes for the RHipe job to finish as well as the load on the hosts and VM's during the test. The average of the 2 PBR scores is used as the final score.

   Viewpoint

   End user and system administrator

2.5.3.6 Test focus areas

The following table represents the focus area for the different tests:

	Bare-metal baseline	VM baseline	Systems administrator	End user
Load test 1	X			
Load test 2			X	X
Load test 3		X	X	X
Load test 4			X	X
Load test 5				X
Load test 6				X
Network test 1	X			
Network test 2		X		
Network test 3				X
Web server test			X	X
Statistical analysis test			X	X

2.5.3.7 Test methodology and plan

The goal of the testing framework is to do load, stability and real-world testing on PCMP's in order for us to populate the metrics of the PCEF. The evaluation of the platforms entailed the following:

1. The selection of platforms to evaluate the PCEF with.

   The following Open Source PCMP's were selected to test the feasibility of the PCEF. The criteria used for the selection was the size and activity of the open source community involved with the projects:

   - Eucalyptus [10].
   - OpenStack [22].
   - OpenNebula [21].

2. Installation of the platform.

3. Customizing (contextualization) of the pre-build image's start-up scripts so that it can initialize properly on the different platforms.

4. Evaluation of the management capabilities as described in Sections 2.5.3.1 and 2.5.3.1.

5. Execute all the load tests in Section 2.5.3.3 using:

- bonnie++.
- LTP.
- netperf.
- Phoronix.

6. Execute real-world use-case tests using:
    - RHipe.
    - Apache web servers with Apache benchmark as the client emulator.
7. Removal of nodes with active VM's on them to test recovering mechanisms.

Sample size for tests The test sample size for all tests were at least three. Constant monitoring and interpretation of results showed only minor fluctuations in the results making it pointless to increase the sample size.

2.5.4 Performance and usage monitoring:

The following well known and widely used monitoring tools were selected for monitoring purposes:

- **Ganglia [44]:** An open-source project that originates from a University of California, Berkeley project. It is targeted at federations of clusters and is widely used to monitor high performance systems such as grids and clusters.

  Usage

  Monitoring and data collection

- **top [45]:** A commonly used Linux program that display system metrics like CPU, memory, disk usage and process activity in real-time.

  Usage

  Monitoring only

- **ntop [45]:** A Linux network traffic tool that shows the network usage in real-time. It comes with a build-in web interface that can be used to monitor the network traffic and hosts on the network. It was installed in the test VM to monitor network traffic in order to determine if subnets were isolated appropriately.

The tool should however be used with caution as the overhead introduced by it is a significant ± 25%, that will skew test results.

Usage

Monitoring only

- **iotop:** A top like tool displaying I/O bandwidth used by processes.

  Usage

  Monitoring only

- **jnettop [46]:** Another top like tool displaying network usage based on host and protocol. *Usage*

  Monitoring only

- **sar [45]** (system activity report) and **iostat [45]:** Linux programs used to report on various system metrics like CPU usage, memory usage, system load and network utilization.

  Usage

  Monitoring and data collection

- **collectl [47]:** A utility with graphing capabilities used to collect performance data simultaneously for the CPU's, disks and network. collectl can also send data directly to Ganglia.

  Usage

  Data collection

2.5.5 Evaluation criteria

The SMI model is adopted as is and we therefore only need to focus on the identification and categorization of Key Performance Indicators (KPI's), according to the SMI attributes.

A very detailed set of KPI's, of which the majority is from the 'General Package Evaluation Criteria Template' published by Borysowich[48] and the

'Research Challenges for Enterprise cloud computing' paper [49] was identified. A smaller set, that will be referred to as the 'Critical KPI's', was identified during the load tests by their abilities to expose either shortcomings or differences between platforms.

The detailed set is fairly large and is available in Appendix A. The section below give details of the Critical KPI's only.

2.5.5.1 Critical KPI's

The KPI's are scored using either the Likert scale if it is a qualitative measurement, or the actual values measured if it is a quantitative measurement. The final score for a KPI is calculated using the PBR, Section 2.5.2, mechanism.

The results of the scoring will be recorded in the following PCEF Categories:

2.5.5.2 Agility

Capacity

PCEF viewpoint	PCEF KPI
Management	Oversubscribing of services
Management and system administrator	Multiple tenants per host

Portability

PCEF viewpoint	PCEF KPI
Management and system administrator	Minimal customization to products required

2.5.5.3 Assurance of Service

Service Stability

PCEF viewpoint	PCEF KPI
Systems administrator and end user	High Availability - maintenance and failures should be transparent to users - self-healing features
Systems administrator	Cloud controller is not a single point of failure
End user	VM is stable in a stressed environment

Reliability

PCEF viewpoint	PCEF KPI
Systems administrator and end user	DNS / DHCP Services function properly
Systems administrator	The overhead of the cloud controller on the system as a whole is acceptable
Systems administrator and end user	VM is healthy after start-up
Systems administrator	System status is persisted to disk (Restart of cloud controller does not affect any worker nodes)

2.5.5.4 Usability

Operability

PCEF viewpoint	PCEF KPI
Systems administrator	The system administration capabilities of the PCMP is adequate for a production environment
Systems administrator and end user	A charge-back model that measures and reports on the end user's resource utilization
Systems administrator and end user	An inventory system that keeps track of VM's issued, licenses issued, resources such as disk, memory, CPU's etc. with trigger points to assist in capacity management
Systems administrator and end user	Operational management system to monitor and diagnose the data center

Suitability

PCEF viewpoint	PCEF KPI
End user	The self-service capabilities of the cloud platform is adequate for a production environment
End user	A service catalogue feature with pre-configured server definitions that allows users to easily select a server and add a custom list of additional services such as database servers, web servers, etc.
End user	Computing resources can be dynamically scaled
End user	A self-service portal - users must be able to manage virtual resources as needed

Installability

PCEF viewpoint	PCEF KPI
System administrator	Installation is straight forward and repeatable
System administrator	VM access configuration is straight forward
System administrator	Automatic discovery of extra worker nodes
System administrator	Automation of human tasks, spread across multiple functional areas, such as VM provisioning, software and platform installs, patch management, security, network and storage configuration, firewall configuration, decommissioning of VM instances, etc. should be orchestrated via a work-flow mechanism

Understandability

PCEF viewpoint	PCEF KPI
System administrator	System documentation available

2.5.5.5 Performance

Service Response Time

PCEF viewpoint	PCEF KPI
End user	Automated processes ensures fast service response times (VM requests and decommissioning, starting and stopping of VM's etc.
End user	VM startup compares favorably with that of Amazon EC2

Suitability

PCEF viewpoint	PCEF KPI
End user and system administrator	Load test percentage based ranking scores

2.5.5.6 Security and Privacy

Access Control and Privilege Management

PCEF viewpoint	PCEF KPI
Management and system administrator	Audit and monitoring

2.6 Populating the PCEF

The scoring for the hierarchical structure defined by the PCEF is started by assigning priorities, a value between 0 and 1, to the different elements in the hierarchy, ensuring that the sum of the priorities for all of the siblings of a node, always equates to one. These priorities represent the relative weights of the nodes (the siblings) to that of their parent.

The steps involved in populating the PCEF are:

- Select the Viewpoints, Categories, Attributes and KPI's that is involved in the evaluation.
- Select the load and real-world tests needed to gather data for the relevant KPI's selected. Use the table in Section 2.5.3.2 as a guide for the selection process.
- Execute the tests using Section 2.5.3.7 as a guide.
- Assign priorities to all elements involved in the evaluation:
  - Assign priorities to the each of the viewpoints ensuring they add up to one.
  - Assign priorities to the 7 top level Categories ensuring they add up to one.

- Assign priorities to each of the attributes associated with a category ensuring the sum of the attributes' priorities for each category equals one.
- Assign priorities to the KPI's associated with each attribute ensuring their sum equals one.

• Score the platforms using the percentage based ranking mechanism on a KPI level using the formulas in section 2.5.2:

- Assign an expected and maximum value to the KPI.
- Calculate the final score for the KPI, for each system, as described in Section 2.5.2.
- The contribution of a KPI to the final score of a system is obtained by multiplying the normalized score with the priorities of the KPI, Attribute, Category and finally, that of the relevant Viewpoint found as you move up the tree.

  For each system's score: Multiply the score with the priority of the KPI Repeat this step for all the KPI's associated with an Attribute and aggregate the results for each system. Multiply the aggregated result with the priority associated with the Attribute to calculate the system's score for the specific Attribute
- Repeat the previous step for all the Attributes associated with a Category, aggregating the results on a system level as you proceed. Multiply the aggregated result with the priority of a Category.
- Repeat the previous step for all the Categories associated with a viewpoint, aggregating the results on a system level as you proceed. Multiply the aggregated result with the priority of the viewpoint.
- Aggregate the results of the viewpoints to get the final score for a system.

  Use this score to determine the ranking order as well as the suitability percentage for a system.

2.6.1 Example

For the example in Figure 2.5 the KPI's contribution to the final score for *System 1* is calculated as follow:

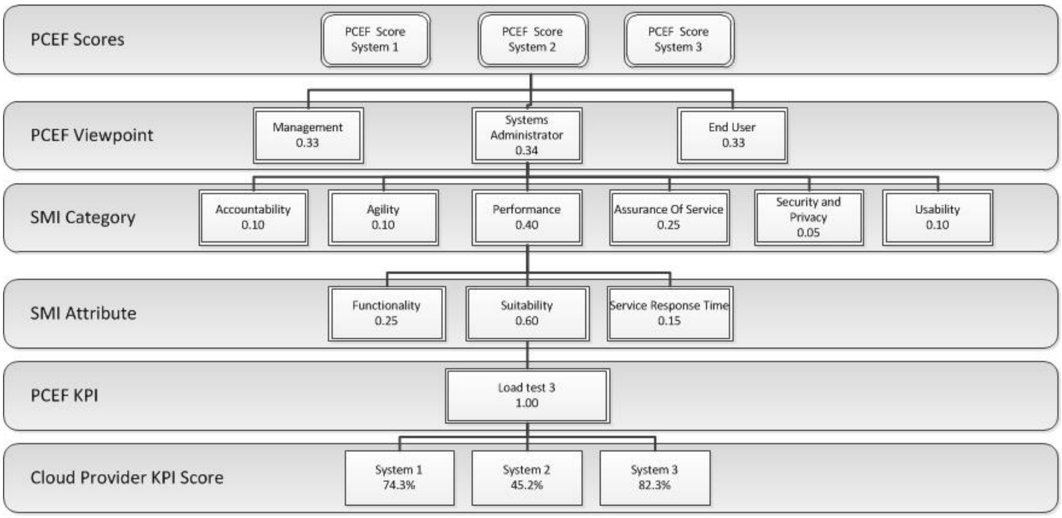


Figure 2.5: PCEF example

PCEF Level	Metric	Score	X	Priority	=	Adjusted Value
KPI	Load test 3	74.30	X	1.00	=	74.30
Attribute	Suitability	74.30	X	0.60	=	44.58
Category	Performance	44.58	X	0.40	=	17.83
Viewpoint	System administration	17.83	X	0.34	=	6.06

The KPI's 74.3% PBR value in the 'System administration viewpoint' will thus contribute 6.06% to the final score for *System 1*

Below is a table showing the calculation for the three platforms in the example:

		Priority				Final
	Score	KPI	Attribute	Category	Viewpoint	Score
System 1	74.30	1.00	0.60	0.40	0.34	6.06
System 2	45.20	1.00	0.60	0.40	0.34	3.69
System 3	82.30	1.00	0.60	0.40	0.34	6.72

2.7 Experimental environment

2.7.1 System configuration

Hardware: The test system was composed of an Intel x86 cluster consisting of six HP Compact 6000 Pro SFF workstations, each equipped with the following hardware:

Dual-core CPU GHz	memory (GB)	Disk (GB)	Network (Mbs)	File System
2.60	4	320	1000	ext4

Topology: One of the machines was reserved as a Test controller leaving 5 machines available for the PCMP hosting nodes. The network was configured as a private LAN using a HP Procurve Switch, with access to the outside world via a second network card on the test controller.

The server farm is sufficiently sized to fulfil the needs of the primary objective of this thesis (the formation of an evaluation framework), but the configuration can also be used for capacity planning purposes during the evaluation phase of a private cloud initiative. It is small enough to justify the expenditure and the 5 hosting nodes is suitable for scaling experiments.

A schematic of the topology can be seen in *Figure 2.6*

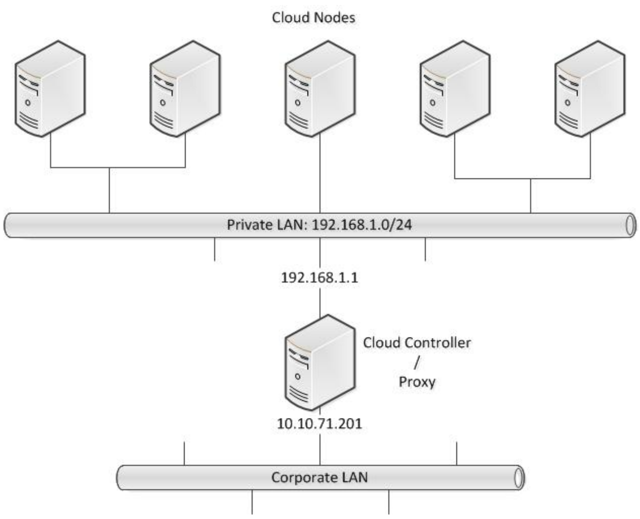


Figure 2.6: Test system topology

2.7.1.1 Test controller configuration

Software: The test controller used Ubuntu 11.10 desktop edition for the operating system with the following additional packages:

Package	Reason
squid3	Used as a proxy to control Internet access from the cloud nodes
ntp	Time server
cacti	Profiling
ntop	Network profiling and monitor
ganglia	Cluster monitoring
euca2ools	Eucalyptus management

2.7.2 Virtual machine configurations used

Two types of VM configurations were used during the tests. The *medium* configuration allowed for 2 VM's to be started on a single host to allow for multi-tenant tests on a single host, while the *large* configuration allowed for tests utilizing as much as possible resources from a single host, without starving the host itself from resources; starving the host itself does not make sense as it by default will give for a poor performance on the VM's.

Table 2.2: VM specifications

Name	CPU cores	Memory(MB)	Disk (GB)
medium	1	1024	40
large	2	2048	40

The comparative tests on Amazon used the following Amazon VM:

Table 2.3: Amazon EC2 VM specification

Name	CPU cores	Memory(MB)	Disk (GB)
m1.large	2	2048	8

2.8 Summary

The chapter kicked off with a brief overview of the related work used as the foundation for our work, and proceeded with the details of the research method followed.

The first step was the definition of a reference model which resulted in further research on:

- A metrics framework with KPI's that can be used to score PCMP's with.
- A diagnostics and monitoring framework to monitor the platforms with while the experiments is running.
- A test suite that can thoroughly test PCMP's.

We also felt it necessary to master virtualization as it forms the basis for cloud computing. This came in handy during the evaluation of the PCMP's because the platforms tested had limited VM management capabilities.

The next phase was the selection of real life use cases as we felt that although our test suite was comprehensive, it might not give a true reflection of a platform as it focuses on load related behavior only. This requirement resulted in research into cloud specific technologies, such as Hadoop, that came with a steep learning curve as it does not follow traditional programming practices. Configuration of these technologies also took a while to figure out because of often incomplete documentation.

The rest of the research involved the execution of experiments which was driven by the test plans compiled in the test suite.

Note: Cloud computing is only now, 2013, starting to gain critical mass, which explains why most of the literature research involved information from the Internet.

Chapter 3

Systems Evaluated

3.1 Introduction

To evaluate the effectiveness of the PCEF an experimental configuration where the behavior of the PCMP's is close to each other is required. This will enable us to test whether the PCEF is capable of identifying differences between platforms. Because of time constraints on the selection of PCMP's, we decided to take the open source route as we could use the activity and feedback of the communities involved as a yardstick to fast track the selection process.

The PCMP's selected were OpenNebula, Eucalyptus and OpenStack as this seemed to be the most active x86 based open source projects at the time of the evaluation. The other advantage is that they all offer a IaaS PCMP which supports KVM, making it perfect fit for the requirements as stated above.

3.2 VM selection

Younge *et al.*[50] did research on the virtualization technologies currently available for cloud computing platforms. They evaluated KVM, Xen and VirtualBox and concluded that hypervisors are a viable option to use in high performance computing systems and that KVM is the best overall choice to use because of its feature-rich experience and near-native performance (They disqualified VMWare due to its limited and costly licensing). We are comfortable with their research approach and result, and will therefor only focus on KVM for our tests.

KVM contains virtualization extensions for Intel VT [51] and AMD-V [52] CPU's and allows you to run multiple virtual machines (VM's) with unmodified Linux or Windows images on the same host. It is part of the Linux kernel that converts the kernel into a bare-metal hypervisor.

3.3 Cloud Management Platform(CMP) Classification

Llorente[5] did a feature based classification of open source platforms and created a chart, the CMP Quadrant, to aid in the understanding of cloud models. The two dimensions of the CMP Quadrant represent the cloud model on the vertical axis and cloud flexibility on the horizontal axis.

The cloud model axis focus on the two extreme models available:

- The Amazon Web Services (AWS) model which is focused on infrastructure provisioning by end users.
- The data center virtualization model which focuses on the management of virtual resources by system administrators.

The flexibility axis represents the ability for the product to adapt to data center services, and the ability to be customized in order to provide a differentiated cloud service.

The following table illustrates some of the differences between the two philosophies:

	Data Center Virtualization	Infrastructure Provision
User	System administrators	End users
Interfaces	Feature-rich API and administration portal	Simple cloud APIs and self-service portal
Management Capabilities	Complete life-cycle management of virtual and physical resources	Simplified life-cycle management of virtual resources with abstraction of the underlying infrastructure
Internal Design	Bottom-up design dictated by the management of data center complexity	Top-down design dictated by the efficient implementation of cloud interfaces
Applications	Traditional multi-tiered enterprise applications	Re-architectured to fit into the cloud paradigm
Enterprise Capabilities	High availability, fault tolerance, replication, scheduling, etc. provided by the CMP	Most of them built into the application, as in 'design for failure'
Data Center Integration	Easy to adapt to fit into any existing infrastructure environment to leverage IT investments	Built on new, homogeneous commodity infrastructure

Figure 3.1 visualize where our PCMP's fits into the classification:

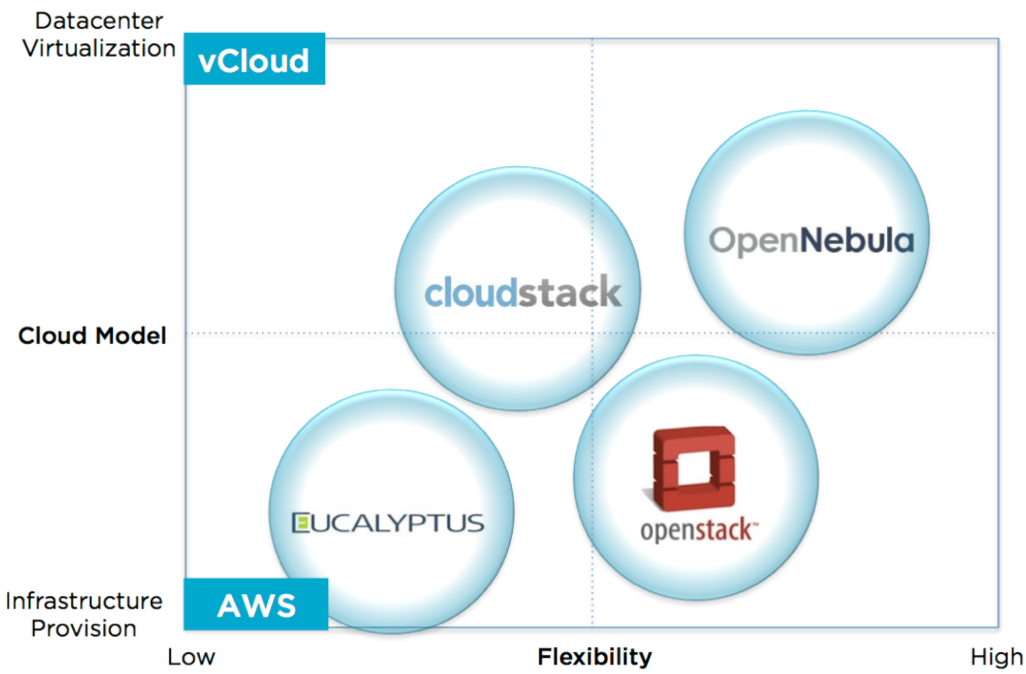


Figure 3.1: CMP Quadrant [5]

Llorente's classification on the platforms we chose shows differences in their philosophies, with Eucalyptus and OpenStack leaning towards the 'infrastructure provision (Amazon)' model and OpenNebula more towards the 'data center virtualization (vCloud)' model.

We don't agree with this point of view and would place all the PCMP's evaluated into the data center virtualization category because of the limited self service capabilities available.

3.4 High level comparison of platforms tested

OpenNebula [21] allows you to build and manage private clouds with Xen, KVM, VMWare ESX, and hybrid clouds with Amazon EC2 and other providers. Although its tool-set seems to receive less attention than some of the other cloud tools, it is used to manage clouds in a number of international projects such as a cloud at CERN in Switzerland, with 500 servers, and the D-Grid Resource Center Ruhr where it manages a cloud infrastructure with several hundred servers.

Eucalyptus [10] uses the Amazon EC2 specification for its Infrastructure as a Service (IaaS) offering and supports most of the Amazon Elastic Compute cloud (EC2) application programming interface (API). It is however not possible to deploy a Eucalyptus VM on Amazon without a migration exercise because of contextualization differences.

OpenStack [22] was started by Rackspace and NASA and have received a lot of international attention resulting in more than 160 companies, including Dell, Cisco, Hewlett-Packard, LP, IBM, Intel, Red Hat and VMware, joining the initiative since its inception in 2010.

Feature Comparison

	Eucalyptus	OpenStack	OpenNebula
Managed By	Eucalyptus Systems	OpenStack Foundation	C12 Labs
License	GPL v3	Apache 2.0	Apache 2.0
AWS Compatible	Yes	Yes	Yes
Hypervisors	Xen, KVM, VMWare	Xen, KVM, VMWare	Xen, KVM, VMWare
Programming Languages	Java, C	Python	C, C++, Java, Ruby
Commercial Support	X	X	X
Cloud model	Data center virtualization	Data center virtualization	Data center virtualization

3.5 Summary

This chapter explained the reasoning behind our choice of PCMP's and hypervisor. We also gave a feature based classification for PCMP's, described where the platforms selected to test the PCEF with fit into the classification, and gave a high level overview of the features of these platforms.

Chapter 4

Results: Systems administrator perspective

4.1 Introduction

This chapter gives an overview of the test results, administration and load tests, from a cloud system administrator's perspective. It covers the system administrator assessment, behavior of the PCMP cloud controllers, a short discussion on VM behavior, the result of the load tests, and conclude with an analysis of the test results.

Note: The *(subscription in Italics)* accompanying each KPI description in the scores below gives the PCEF category and attribute for the KPI.

4.2 Systems administration capabilities

The objective of the system administrator assessment was to determine whether the tools provided by the PCMP's were sufficient to satisfy the operational requirements needed by system administrators.

The following tests were performed in addition to the standard system administration tasks to identify discrepancies between platforms:

- Restart of cloud controller:

  This test was done with multiple VM's deployed to test:

  - Whether a controller restart would affect the VM's.

  - If the controller persist the state of the system.

  The health of all system administration commands were also evaluated after the restart.

- Restart of hosting nodes.

  This test was done with zero or multiple VM's deployed on a host to test whether the cloud controller would:

  - Notice the reboot and update its registry, and notify the owners of deployed VM's if they were affected.
  - Automatically restart a lost VM on another node.

The sections to follow describe the system administration experience and give summaries of issues encountered.

4.2.1 Installation

OpenNebula OpenNebula was by far the easiest to install as it only needed a small, ± 10MB, install on the cloud controller machine. The only requirement for the hosting machines is ssh access, ruby and a working hypervisor.

Eucalyptus There were quite a few issues with the Eucalyptus installations distributed with the Ubuntu Enterprise Cloud (UEC) that caused a number of failed installation attempts. The software seems to behave differently when different country regions are selected for example. The most stable version seems to be when the defaults, USA, are selected at every option during the installation. Other issues such as software updates also seem problematic and needs tinkering to get it to work.

OpenStack OpenStack installed without incident, but getting the platform to work was complex because of incomplete and sometimes contradicting, installation instructions and security issues.

General Most of the issues experienced during the installation of the cloud platforms aligns with the findings of Pelletingeas[53]. This was not unexpected however and the installation was used for up skilling purposes. It should be a straight forward exercise if proper training was done beforehand.

Scores

KPI	OpenNebula	Eucalyptus	OpenStack
Installation is straight forward and repeatable. *(Usability -> Installability)*	4	1	3
Minimal customization to get VM's to function in the PCMP is required. *(Agility -> Portability)*	4	4	4

4.2.2 Management

OpenNebula The Graphical User Interface (GUI) used for management of the OpenNebula platform offered the most features of the PCMP's evaluated but was still not as feature rich as one would expect to see in a production environment.

Eucalyptus The graphical user interface provided with Eucalyptus is very basic and does not provide any useful functionality that can be used in a production environment. All of the administration functionality needed for VM deployment and management is done from the command line. The management however lacks critical functionality for a production environment such as an indication on where a VM was deployed. This makes debugging of the system extremely cumbersome and is not acceptable from a system administrator's point of view.

The following issues were encountered during the evaluation:

- Eucalyptus only support ext2 and ext3 file systems.
- Nodes that was added are not visible if a DNS was not specified during the base installation.
- Running images are not accessible (IP and ICMP) from the cloud controller if the Image IP's are in the same subnet as that of the cloud system.

OpenStack The OpenStack GUI was not at the same level of that of OpenNebula but a definite improvement on the Eucalyptus attempt.

Scores

KPI	OpenNebula	Eucalyptus	OpenStack
The PCMP provide an Audit and Operational Monitoring solution out of the box. *(Security and Privacy -> Access Control and Privilege Management)*	4	2	3
The PCMP provides a reporting system that can be used for consumption tracking and billing purposes *(Usability -> Suitability)*	3	1	2
The system administration and operational management capabilities of the PCMP is adequate for a production environment *(Usability -> Operability)*	4	1	3
The PCMP provides a chargeback model that measures and reports resource utilization *(Usability -> Operability)*	1	1	1
The PCMP provides an inventory system that keeps track of VM's and licences issued as well as resource usage with trigger points to assist in capacity management *(Usability -> Operability)*	4	3	3

Table 4.2 – *Continued from previous page*

KPI	OpenNebula	Eucalyptus	OpenStack
Maintenance and failures are transparent to users via autonomous self-healing capabilities offered by the PCMP *(Assurance of Service -> Service Stability)*	2	2	2

4.3 Cloud controller(CC) behavior

4.3.1 Cloud controller overhead on hosting nodes

OpenNebula The CC for OpenNebula caused no significant load on idle and active nodes and behaved reliably during the tests.

Eucalyptus The CC for Eucalyptus caused no significant load on idle and active nodes but it caused the following critical failures:

Basic functionality issues on Eucalyptus:

- A restart of the cluster cause 'euca-describe-availability-zones' to report no resources available.
- The CC does not always restart properly after a reboot and have to be restarted manually.
- The DNS functionality provided with Eucalyptus is buggy and don't function properly with reverse lookups. The query is executed against the public IP address but the reply is send via the internal IP causing chaos with the configuration of software, like RHipe for example, which depends on DNS lookups. A work around was implemented by adding all the hostnames and IP addresses to the local 'hosts' files of the nodes.

OpenStack The OpenStack CC sometimes had an average load (measured over a 15 minute interval) of $\pm$ 0.8 on the hosting nodes with one idle VM deployed. This seems a bit high when compared to the other PCMP's where almost no load on the hosting nodes were measured . This was the only concern and the system behaved reliable during the tests.

Scores

KPI	OpenNebula	Eucalyptus	OpenStack
The overhead of the PCMP on the system as a whole is acceptable *(Assurance of Service -> Reliability)*	5	5	4
The system status is persisted to disk resulting in no service interruption if the cloud controller is restarted *(Assurance of Service -> Reliability)*	4	1	3

4.4 VM behavior

OpenNebula There was no issues recorded with VM's during the OpenNebula assessment.

Eucalyptus The CC handles the startup of VM instances poorly and the following issues were recorded during VM launches:

- The hosting nodes, on which VM's are deployed on, sometimes become completely unresponsive (usually during the load tests) resulting in a restart of the node.
- The contextualization script, rc.local, sometimes fail to execute properly causing host names and certificate loading to fail, resulting in the VM to be of no use as you cannot access it.

- After multiple VM launches and load tests, nodes sometimes end up in a state where VM's can't be started. This can be resolved by cleaning the PCMP VM cache followed with a restart of the CC.
- An idle Apache daemon on a newly deployed VM would consume 100% of a CPU with Eucalyptus 1.6 and was causing an average load of 1.2 on the CC. This behavior was not experienced with any other PCMP and we concluded the CC to be at fault.

OpenStack:

- After a reboot of the cloud controller, a newly deployed VM will fail to retrieve its meta-data which include the key needed to ssh into the instance (similar to the Eucalyptus issue). You will thus not be able to ssh to the VM. The resolution for this issue was to uninstall the nova-api, reboot and re-install the nova-api.
- There is a runaway process, mountall, on the nodes after startup that consumes 100% of a CPU. The only way to get rid of the process was to forcefully (kill -9) kill it after the restart.

Overcommitting of CPU's and memory Overcommitting is the allocation of more virtual resources than the amount of physical resources available.

None of the platforms evaluated could deploy more VM's than the physical number of CPU cores available, despite claims by the KVM hypervisor that it support the overcommitting of CPU's and memory [54]. This feature is necessary as there is bound to be idle VM's deployed on a host. It was identified as a risk and is covered by the 'The PCMP support the overcommitting of CPU's and memory' KPI.

Scores

KPI	OpenNebula	Eucalyptus	OpenStack
The PCMP support the overcommitting of CPU's and memory *(Usability -> Suitability)*	1	1	1

Table 4.4 – *Continued from previous page*

KPI	OpenNebula	Eucalyptus	OpenStack
PCMP support multiple tenants per host. *(Agility -> Capacity)*	4	3	3
VM's are healthy and accessible after commissioning *(Assurance of Service -> Reliability)*	4	1	2
The DNS/DHCP Services function properly *(Assurance of Service -> Reliability)*	4	1	4

4.5 Load and real-world tests

This section discusses the load and real-world tests that were executed against the PCMP's. The results will be combined to calculate the PBR for the platforms and will be recorded in the *(Performance -> Suitability)* section of the PCEF.

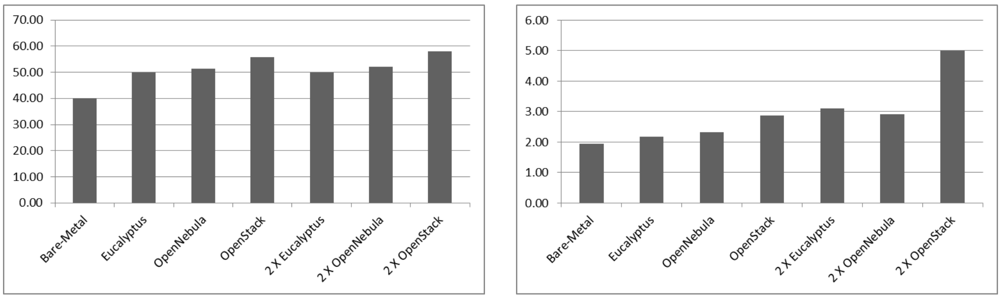
4.5.1 Load test 1

The objective of Load test 1 is to capture a bare-metal baseline result set that can be used by the other tests as the expected value with the PBR scoring.

4.5.2 Load test 2

The primary objective, from a system administration viewpoint, of the test was to test the effect of deployed VM's on a server. The deployed VM was in an idle state with no active applications apart from the OS and idle daemons.

The graphs below show the CPU utilization and CPU load for the host on which the VM was deployed. The bare-metal counts are the results of Load test 1. The first set of PCMP counts show the effect on the host with one idle VM deployed on it, and the next set shows the effect when there are two VM's deployed per host. The low VM count is because of the lack of CPU overcommitting support by the PCMP's evaluated, as discussed in Section 4.4.



Host CPU utilization Host CPU load

Load test 2: Idle VM effect on host

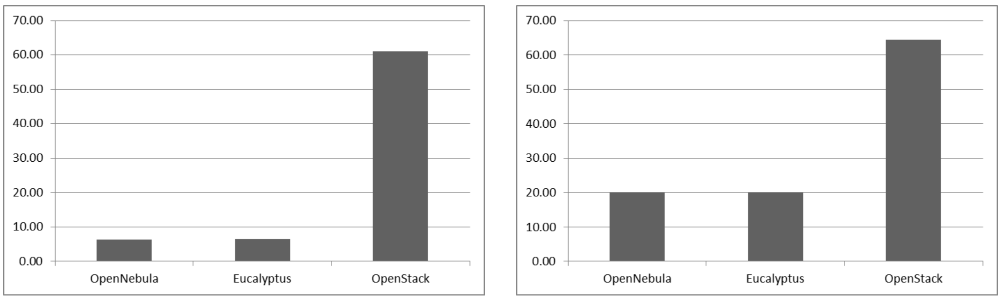
The above graphs clearly show OpenStack's increased load, compared to that of the other platforms, when more than one VM is deployed.

4.5.3 Load test 3

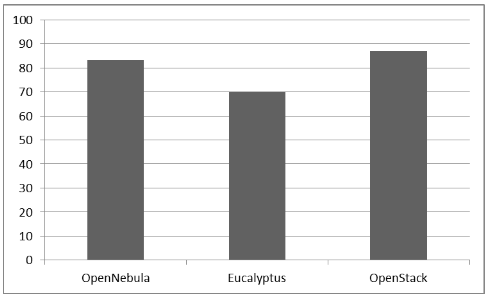
The objective of this test from a system administrator point of view is to measure the impact of a stressed VM on a host. For this experiment bonnie++, LTP, and the Phoronix Test Suite were executed on the VM and the CPU load on the hosts were recorded.

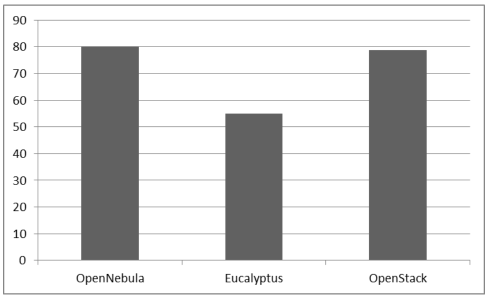
The graphs below shows the CPU utilization and CPU load for the different tests performed.

VM induced CPU utilization on host

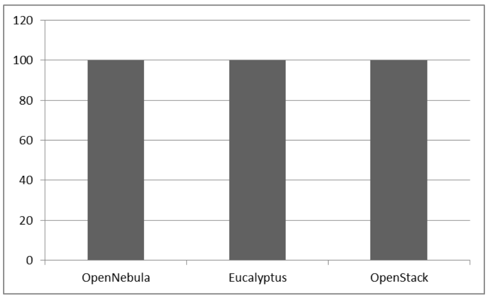


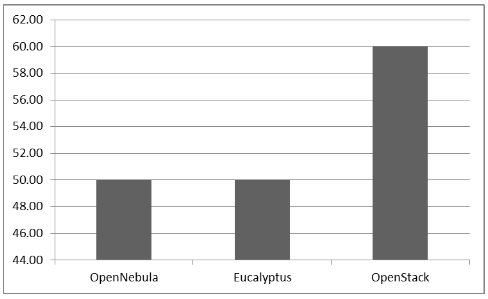
bonnie++ LTP





7-zip g-zip

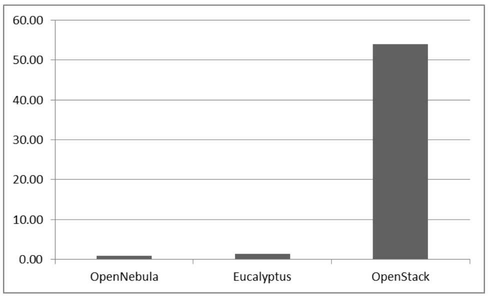


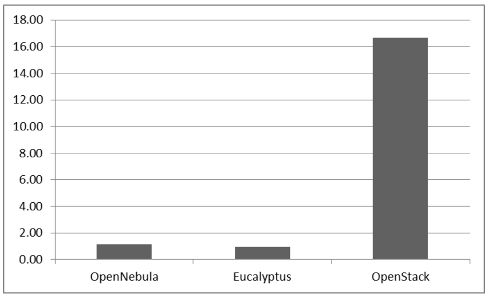


C-Ray memory

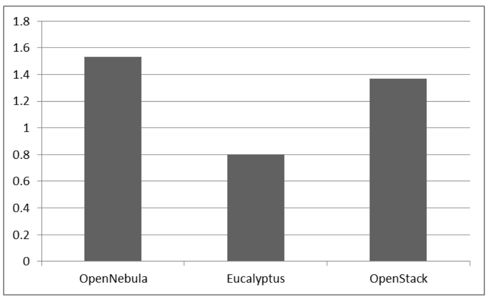
Load test 3: VM induced CPU utilization on host

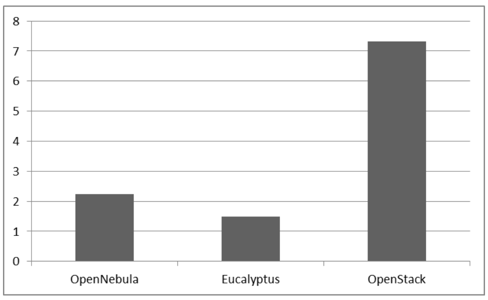
VM induced CPU load on host



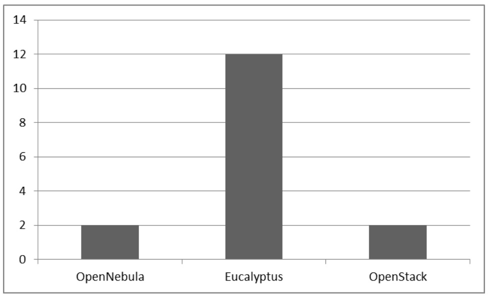


bonnie++ LTP

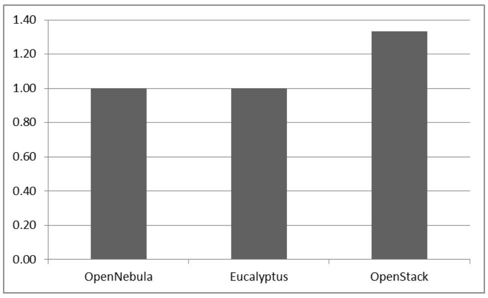




7-zip



g-zip



C-Ray memory

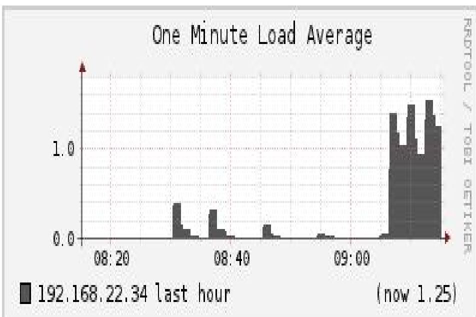
Load test 3: VM induced CPU load on host

The high load experienced by OpenStack during the tests that involved disk I/O is noteworthy. It points to a potential problem for system administrators as they will need more physical resources for a disk intensive system if they use OpenStack.

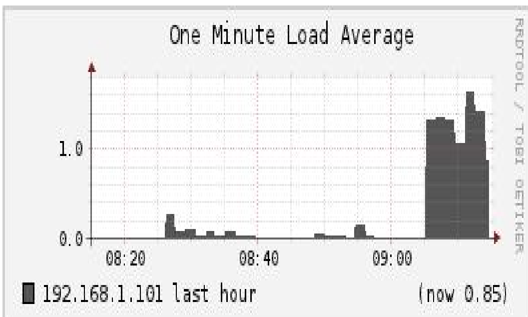
The section below gives a more in-depth description of the OpenStack behavior using the data from one of the g-zip experiments.

4.5.3.1 g-zip

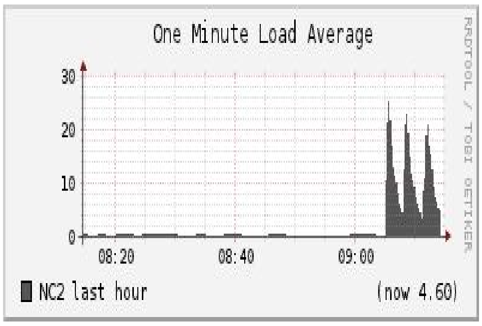
The g-zip test created and compressed a 2GB file, and although the time and load on the VM's for the test was similar across all three PCMP's, the load imposed by the OpenStack VM on the hosting machine was extremely high if compared to that of the other platforms. The graph below shows the negative effect the g-zip file creation activity has on the OpenStack hosting machine.



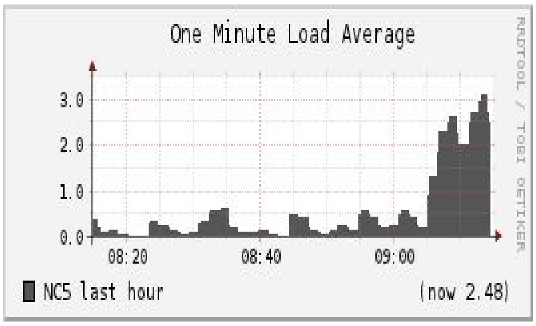
OpenStack VM load



OpenNebula VM load



OpenStack Host load



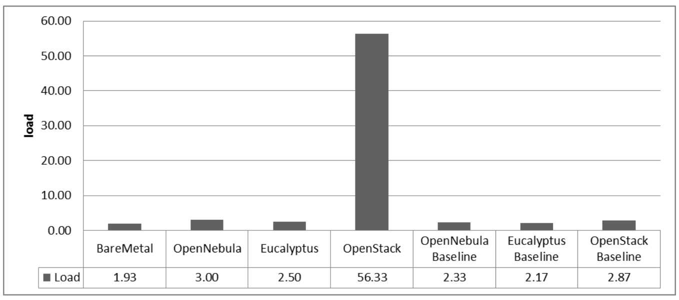
OpenNebula host load

4.5.4 Load test 4

The primary objective of Load test 4 was to test if the PCMP's can handle a stressed environment, and the secondary objective was to test the effect of a stressed host and VM fighting each other for resources.

The graph below gives the results with the bare-metal and Load test 2 (baseline with 1 VM deployed) results included for comparative purposes. There are two interesting observations here:

- OpenStack once again have a surprisingly high load on the hosts.
- The load on the OpenNebula and Eucalyptus hosts are only slightly higher for this test than that of the baseline result.



host load

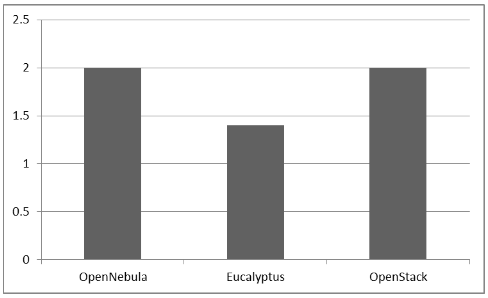
Load test 4: Stressed host and VM fighting for resources

4.5.5 Real-world tests

The purpose of the real-world tests were to evaluate the behavior of the platforms using a test which resembles a real-life situation.

4.5.5.1 RHipe log file analysis

The platforms are ranked by comparing the load on the hosts with each other.

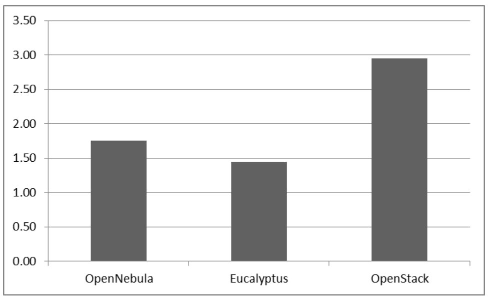


RHipe load

R, the core of RHipe, is an in-memory application and we expected the RHipe results to be in line with the CPU and memory test results. This was indeed the case and we conclude that OpenStack compares favorably with the other platforms for CPU and memory intensive systems but not for disk I/O intensive systems.

4.5.5.2 Web server

The platforms are ranked by comparing the load on the hosts with each other.



Web server load

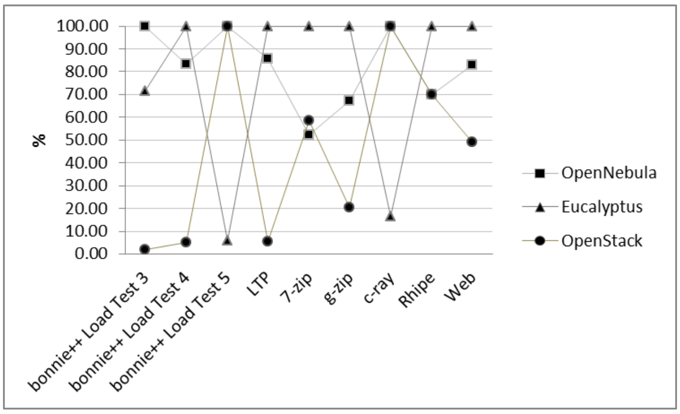
The average of the Apache and Tomcat results are used for the scoring. OpenStack again shows a higher load which we believe is a result of the disk operations involved.

4.6 Performance results analysis

The system administrators main concern from a performance point of view is the load induced on the infrastructure by VM's. This section uses the data collected during the load and real-world tests to score and rank the PCMP's based on the load experienced by the hosting machines.

4.6.1 Percentage based ranking scores

The final scores are calculated by using the results of all the tests, except that of load tests 1 and 2, as we are interested in load related behavior only.



	OpenNebula	Eucalyptus	OpenStack
μ	82.40	77.12	45.62
σ	15.75	36.34	37.23

Percentage based ranking: System administrator perspective

OpenNebula got the best average score and was also the most consistent with the lowest Standard Deviation. The scoring mechanism counted against OpenStack because of its behavior when VM disk I/O is high.

4.6.1.1 Calculation

The following subsection give the details on the scoring calculation.

	Expected Values	Observed Values OpenNebula	Observed Values Eucalyptus	Observed Values OpenStack
bonnie++ Load test 3	1.00	1.00	1.40	54.00
bonnie++ Load test 4	2.50	3.00	2.50	48.00
bonnie++ Load test 5	0.10	0.10	1.67	0.10
LTP	0.97	1.13	0.97	17.50
7-zip	0.80	1.53	0.80	1.37
g-zip	1.50	2.23	1.50	7.33
C-Ray	2.00	2.00	12.00	2.00
RHipe	1.40	2.00	1.40	2.00
Web	1.45	1.75	1.45	2.95

	Normalized Values OpenNebula	Normalized Values Eucalyptus	Normalized Values OpenStack
bonnie++ Load test 3	100.00	71.43	1.85
bonnie++ Load test 4	83.33	100.00	5.21
bonnie++ Load test 5	100.00	5.99	100.00
LTP	85.84	100.00	5.54
7-zip	52.29	100.00	58.39
g-zip	67.26	100.00	20.46
C-Ray	100.00	16.67	100.00
RHipe	70.00	100.00	70.00
Web	82.86	100.00	49.15
μ	82.40	77.12	45.62
σ	15.75	36.34	37.23

4.7 Summary

Eucalyptus consistently scored better than OpenStack but it unfortunately had various critical failures during the tests causing it to be unsuited for a production system.

OpenStack sporadically had a much higher load than the other platforms during the load tests and also had a few critical failures making it less attractive, than OpenNebula for example, to system administrators.

OpenNebula seems to have been the most consistent and stable with an acceptable overall performance making it, from a performance point of view, the first choice for a system administrator.

From both a qualitative and quantitative viewpoint OpenNebula proofed to be the most stable and its overhaul performance was also the best with none of the erratic behavior noticed with OpenStack or critical failures with Eucalyptus. From a system administrator's perspective it would thus be the preferred option because of its performance, stability and predictable behavior.

Chapter 5

Results: End user perspective

5.1 Introduction

This chapter gives an overview of the test results, administration and load tests, from a cloud end user's perspective. It covers the self-service capabilities of the PCMP's, the behavior of the VM's, the result of the load tests, the PCMP overhead on a VM, a comparison of the PCMP's tested on Amazon EC2, and conclude with an analysis of the test results.

Notes:

- A special warning on the monitoring of VM's is necessary:

  The relationship between a virtual guest and the hypervisor might cause some metrics like CPU utilization to be inaccurate for operations such as networking. The reason for this is that some of the processing happens in the hypervisor, causing it to be invisible to monitoring mechanisms in the guest. We will therefore only focus on the overall CPU utilization of a VM, and not on specific CPU utilizations such as that of the network or disk system.

- The *(subscription in Italics)* accompanying each KPI description in the scores below gives the PCEF category and attribute for the KPI.

5.2 Self-service capabilities

All of the platforms tested provided a command line as well as web based user interface for administration and feedback purposes. None of the PCMP's tested had a GUI that was sufficient for a production environment with OpenNebula offering the most features and Eucalyptus almost no functionality at all.

Scores

KPI	OpenNebula	Eucalyptus	OpenStack
The Self-service capabilities of the PCMP provides a work-flow based system to manage virtual resources. *(Usability -> Suitability)*	3	1	2
The PCMP provides a service catalogue via a self-service storefront with pre-configured server definitions that allows users to easily select a server and add a custom list of additional services such as database servers, web servers, etc. *(Usability -> Suitability)*	3	1	2
The PCMP provides a reporting system that can be used for consumption tracking and billing purposes *(Usability -> Suitability)*	3	1	1
The PCMP provides an inventory system that keeps track of VM's and licenses issued as well as resource usage with trigger points to assist in capacity management *(Usability -> Operability)*	3	1	2

Table 5.1 – *Continued from previous page*

KPI	OpenNebula	Eucalyptus	OpenStack
The PCMP provides an operational management system to monitor and diagnose VM's deployed *(Usability -> Operability)*	3	1	2

5.3 Virtual machine behavior

5.3.1 Service start-up and response

OpenNebula OpenNebula uses local caches for the VM images on the hosting nodes which need to be populated the first time a VM is deployed to a node. This causes the first instance launch on a node to always be slow.

Eucalyptus Eucalyptus also use local caches for the VM images on the hosting nodes which need to be populated the first time a VM is deployed to a node. This causes the first instance launch on a node to always be slow.

OpenStack OpenStack seems to prepare the image (including compression) after registration resulting in a much faster launch than the other platforms tested.

Scores

KPI	OpenNebula	Eucalyptus	OpenStack
VM startup compares favorably to that of Amazon EC2 *(Performance -> Service Response Time)*	3	2	4

Table 5.2 – *Continued from previous page*

KPI	OpenNebula	Eucalyptus	OpenStack
Automated processes ensure fast service response times (VM requests and decommissioning, starting and stopping of VM's etc. *(Performance -> Service Response Time)*	3	1	2
VM's are healthy and accessible after commissioning *(Assurance of Service -> Reliability)*	4	1	2
DNS / DHCP Services function properly *(Assurance of Service -> Reliability)*	5	2	3

5.3.2 Stability

OpenNebula There was no issues recorded during the OpenNebula experiments.

Eucalyptus Eucalyptus did not handle a stressed host environment very well and the VM's would occasionally crash during Load test 2, if the ntop service was active on the VM, and Load test 6. This might have catastrophic consequences for example, if backups are run on hosting nodes. This behavior can easily be re-produced by logging onto the VM while a test is active and performing simple tasks like running top or collectl.

OpenStack There was no issues recorded during the OpenNebula experiment.

Scores

KPI	OpenNebula	Eucalyptus	OpenStack
Maintenance and failures are transparent to users via autonomous self-healing capabilities offered by the PCMP *(Assurance of Service -> Service Stability)*	1	1	1
VM's are stable in a stressed environment *(Assurance of Service -> Service Stability)*	4	1	4

5.3.3 Scalability

The dual-VM load tests were used for the scalability evaluation.

OpenNebula There was no issues recorded during the OpenNebula experiment.

Eucalyptus Eucalyptus failed the dual-VM test with the disk sub-system going into a read-only mode causing the VM's to become unresponsive and throwing exceptions, requiring a restart of the VM's for service to be restored.

OpenStack There was no issues recorded during the OpenNebula experiment.

Scores

KPI	OpenNebula	Eucalyptus	OpenStack
Computing resources can be dynamically scaled *(Usability -> Suitability)*	3	1	3

5.4 Load and real-world tests

This Section discusses the load and real-world tests executed against the PCMP's. The results will be combined to calculate the PBR for the platforms and will be recorded in the *(Performance -> Suitability)* section of the PCEF.

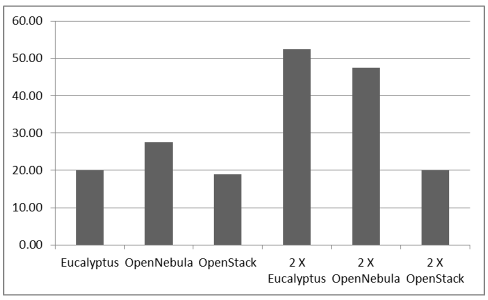
5.4.1 Load test 1

The purpose of Load test 1 is the same as that of the system administrator and was covered in Section 4.5.1.

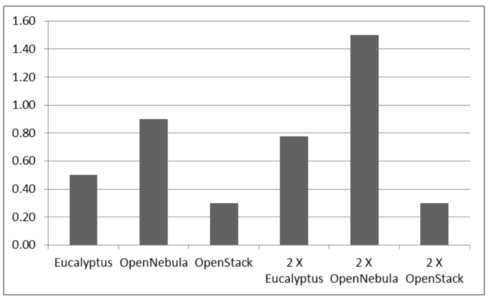
5.4.2 Load test 2

The experiment investigated the effect of a stressed host on an idle VM. Ideally the host should not affect the VM's but this is not realistic as the VM and the host eventually competes for the same system resources. The test can therefore only test for anomalies between the platforms and we expected to see the same VM load across all as they all use the same hypervisor. This was however not the case, as seen in the graphs below, and OpenNebula had an unexceptionably high load compared to that of OpenStack.

The graphs below represent the results and show the CPU utilization and load on the VM for a one and dual VM deployment.



VM CPU utilization



VM CPU load

Load test 2: Stressed host's effect on VM load

The PBR score for the test is calculated by using the results for the dual-VM test (multiple hosts deployed on a VM would be the norm) and the OpenStack value is used as the expected value.

Results

Expected Value	OpenNebula Measured	Eucalyptus Measured	OpenStack Measured
0.3	1.5	0.78	0.3

Scores:

	OpenNebula	Eucalyptus	OpenStack
μ	20.00	38.71	100.00

5.4.3 Load test 3

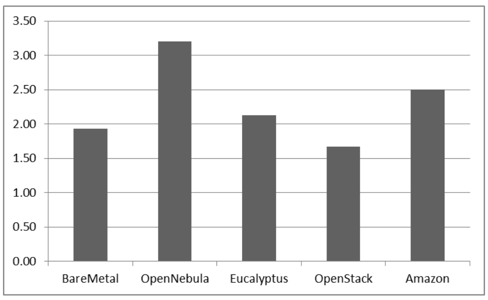
The purpose of this experiment was to stress the VM in order to compare the results of the platforms against each other. This will give us an indication of the effect of the design philosophies of the platforms. Once again the same performance across all platforms was expected because the hypervisor is the same for all of them.

5.4.3.1 Load test 3: bonnie++ results

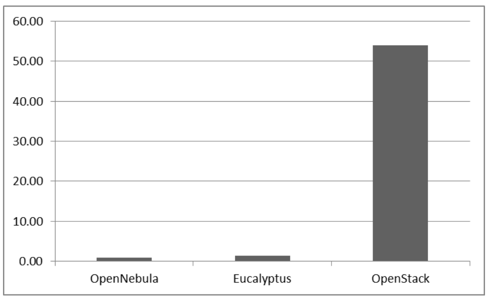
The graphs below show that the platforms did not behave as expected and huge differences were recorded.

OpenNebula and OpenStack are the interesting ones to note as their behavior is exactly the opposite, although not as severe, as the results obtained from the system administrator tests.

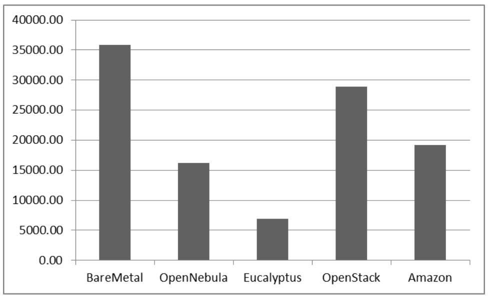
It looks as if the OpenStack hypervisor is configured to optimize disk I/O from a VM point at the expense of the hosting machine, whereas the OpenNebula hypervisor seems to be throttling the disk I/O in order to protect the hosting machine.



VM load



Host load

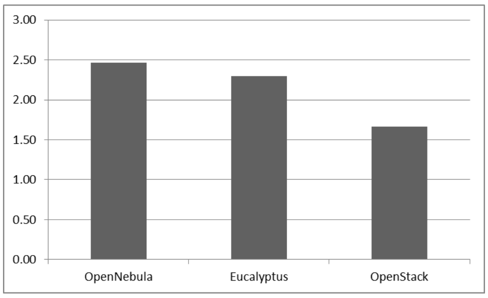


VM average throughput (kB / sec)

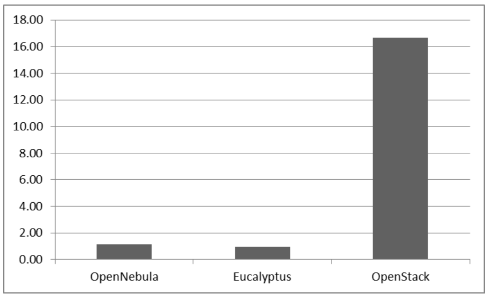
Load test 3: bonnie++ results

5.4.3.2 LTP

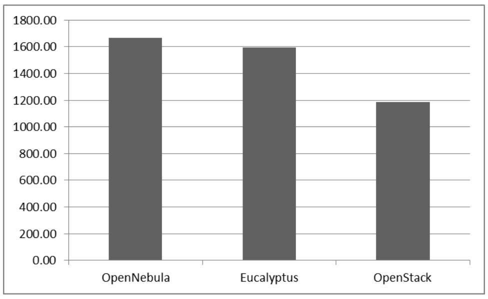
The LTP results had the same pattern as that of bonnie++ with OpenStack taking the lead with both the time-to-execute as well as the CPU load on the VM. The load that the OpenStack VM transferred to the host is however excessive if compared to that of the other PCMP's.



VM load



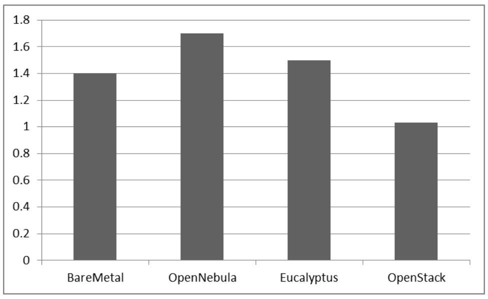
Host load



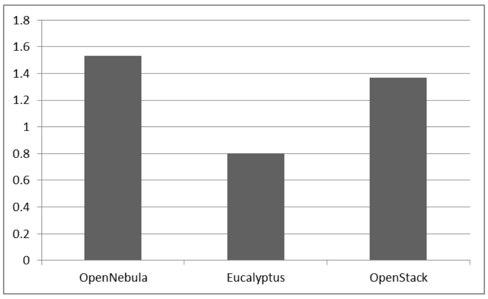
Test execution time (seconds)

5.4.3.3 7-zip

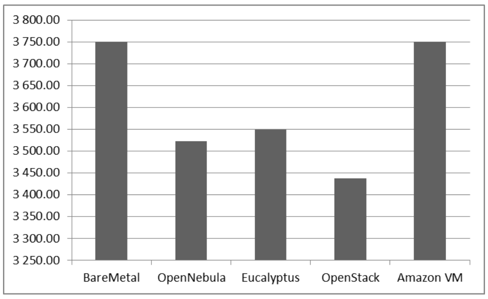
7-zip performed as expected with the results close to each other, but worse than the bare-metal result.



VM load



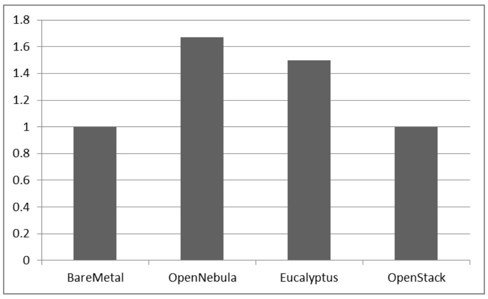
Host load



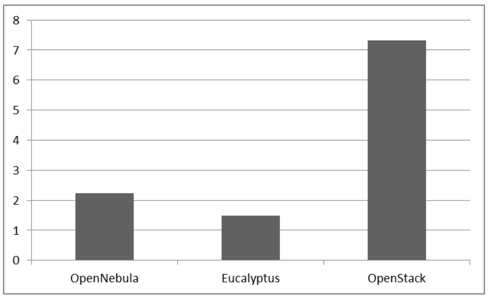
Throughput (MIBS)

5.4.3.4 g-zip

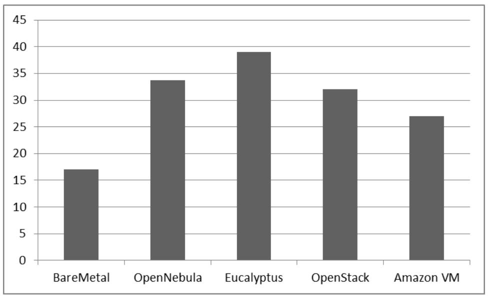
The g-zip test, which includes a disk I/O component, once again showed OpenStack's peculiar host behavior. What is interesting is that although OpenStack caused a higher load on the host than the other platforms evaluated, it did not outperform them.



VM load



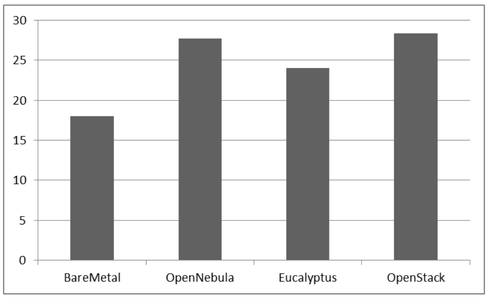
Host load



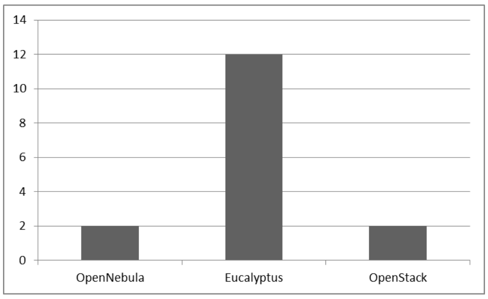
Test execution time (seconds)

5.4.3.5 C-Ray

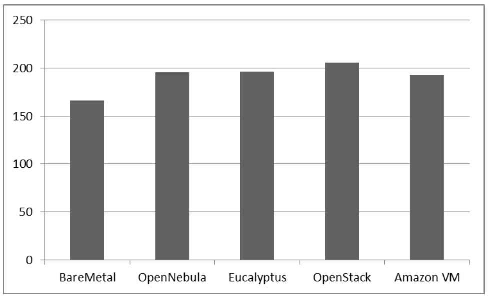
In the C-Ray test it was Eucalyptus that caused a surprise with a high host load that does not make sense. From a VM point of view however everything is fine with all platforms behaving similarly.



VM load



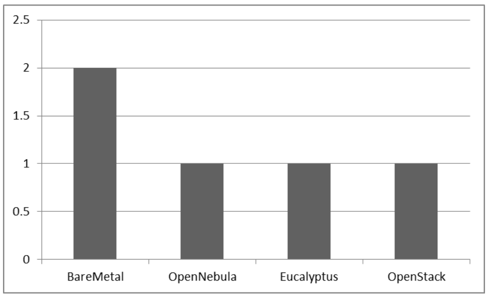
Host load



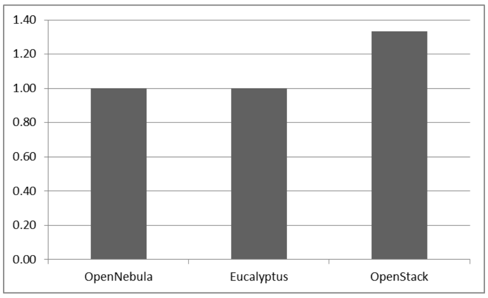
Test execution time (seconds)

5.4.3.6 memory

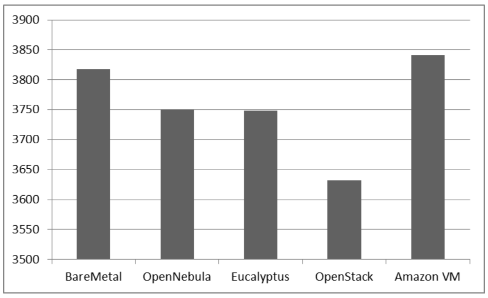
For this test OpenStack had a slightly lower throughput than the rest of the PCMP's tested.



VM load



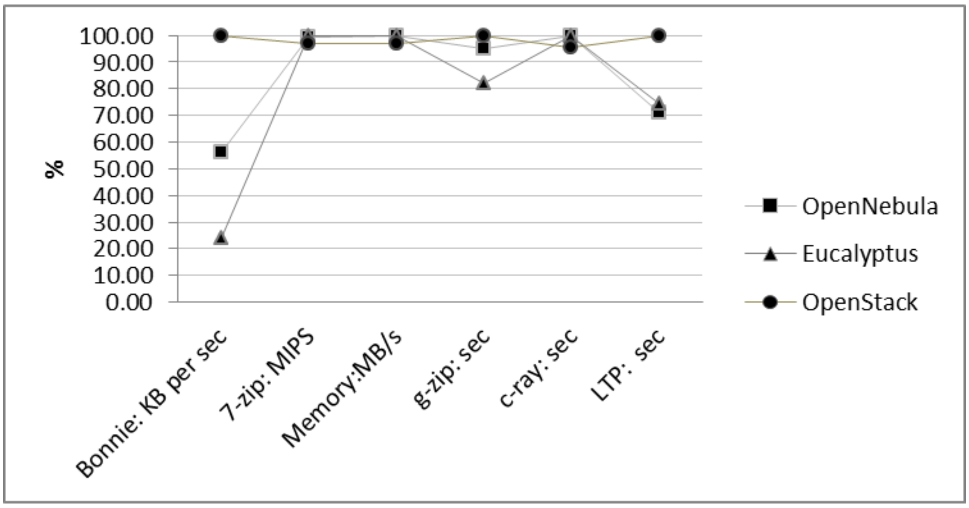
Host load



Throughput MB/s

5.4.3.7 Load test 3 PBR

The PBR score for Load test 3 is calculated by scoring the platforms against each other, using the best result for a test as the expected value.



	OpenNebula	Eucalyptus	OpenStack
μ	86.93	80.00	98.17
σ	17.11	27.00	1.90

Percentage based ranking: Load test 3

Calculation

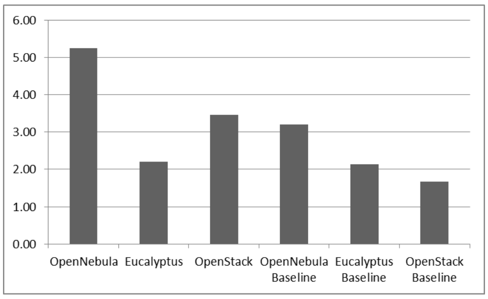
	Expected Values	Observed Values OpenNebula	Observed Values Eucalyptus	Observed Values OpenStack
Bonnie: KB per sec	28952.25	16256.00	6922.92	28952.25
7-zip: MIPS	3549.00	3523.33	3549.00	3437.33
Memory:MB/s	3750.41	3750.41	3748.11	3632.56
g-zip: sec	32.00	33.67	39.00	32.00
C-Ray: sec	195.67	195.67	196.00	205.33
LTP: sec	1184.50	1665.33	1594.33	1184.50

	Normalized Values (maximum=100)		
	OpenNebula	Eucalyptus	OpenStack
Bonnie: KB per sec	56.15	23.91	100.00
7-zip: MIPS	99.28	100.00	96.85
Memory:MB/s	100.00	99.94	96.86
g-zip: sec	95.04	82.05	100.00
C-Ray: sec	100.00	99.83	95.30
LTP: sec	71.13	74.29	100.00
μ	86.93	80.00	98.17
σ	17.11	27.00	1.90

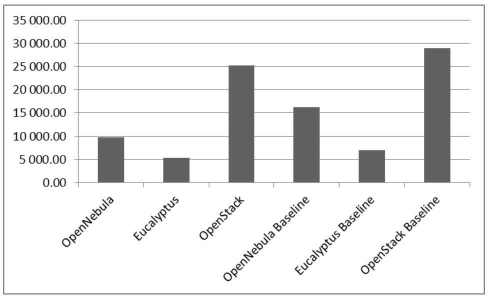
Percentage based ranking results: End user (Load test 3)

5.4.4 Load test 4

The primary objective of the test was to test whether the PCMP's can handle a stressed environment from both a hosting node point of view, as well as from the VM point of view. The secondary objective was to test the effect of a host and VM fighting each other for resources. The platforms are ranked based on the throughput for bonnie++ and using the best baseline value obtained in Load test 3 as the expected value.



VM load



bonnie++ kB per sec

Load test 4: Stressed host and VM competing for resources (end user View)

OpenNebula is clearly the odd one out with it only achieving 50% of the throughput it managed in Load test 3.

PBR Score Calculation

The PBR score for the test is calculated using the throughput results and the OpenStack Baseline result achieved in Load test 3 as the expected value.

Results

Expected Value	OpenNebula Measured	Eucalyptus Measured	OpenStack Measured
28,952.25	9,713.75	5,349.42	25,238.75

Scores:

	OpenNebula	Eucalyptus	OpenStack
μ	33.55	18.48	87.17

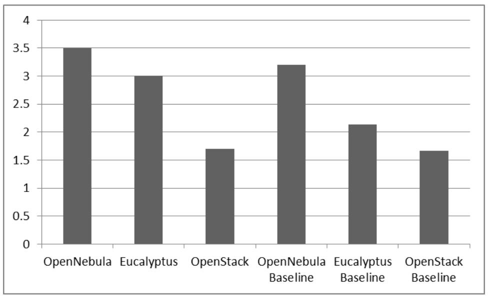
5.4.5 Load test 5

The primary objective of Load test 5 was to test the effect of a stressed VM on co-hosted VM's. The platform rank is based on the load on the idle VM's as well as the bonnie++ throughput. Load test 3's best baseline result is used as the expected value.

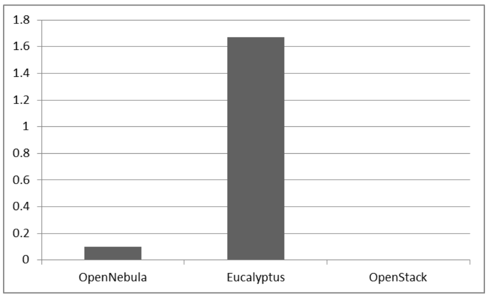
Overall none of the platforms seemed severely affected by the dual VM configuration. The only exception was with Eucalyptus where the second, idle, VM showed a significantly higher load than that for the other platforms.

Calculations

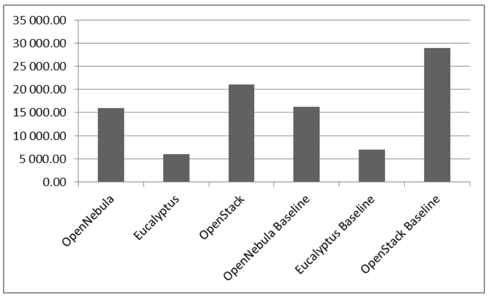
The PBR score for the test is calculated by using the throughput results, and the OpenStack Baseline result achieved in Load test 3 as the expected value.



1st VM load



2nd VM load



bonnie++ kB per sec

Load test 5: Stressed VM with co-located VM in idle state

Results bonnie++

Expected Value	OpenNebula Measured	Eucalyptus Measured	OpenStack Measured
28,952.25	15,973.00	5,968.17	21,104.00

Stressed accompanying VM penalty

Expected Value	OpenNebula Measured	Eucalyptus Measured	OpenStack Measured
0.10	0.10	1.67	0.10

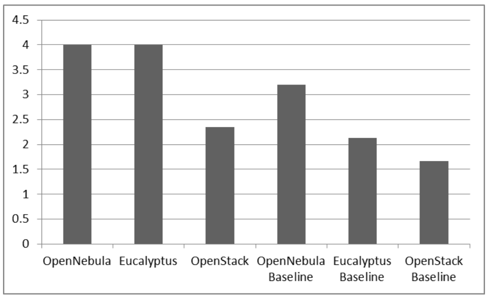
Scores:

	OpenNebula	Eucalyptus	OpenStack
bonnie++ throughput	55.17	20.61	72.89
Stressed accompanying VM penalty	100.00	5.99	100.00
μ	77.58	13.30	86.45

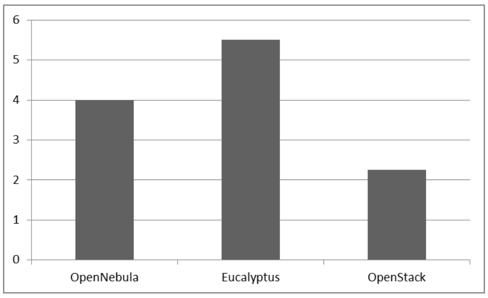
5.4.6 Load test 6

The primary objective of Load test 6 was to test the effect of stressed VM's on each other. The platform's rank is based on:

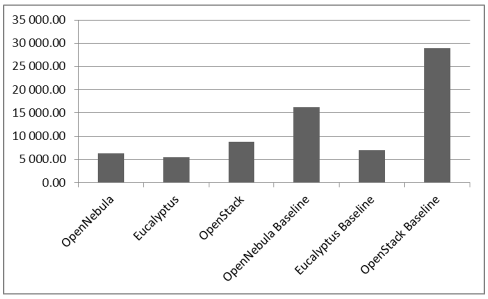
- The load on the VM's, using the best result as the expected value.
- The bonnie++ throughput, using the best baseline result achieved in Load test 3 as the expected value.



VM load



2nd VM load



bonnie++ kB per sec

Load test 6: Stressed VM's competing for resources

Calculations

The PBR score for the test is calculated by using the throughput results for VM 1, and the OpenStack Baseline result as the expected value.

bonnie++ Throughput results

Expected Value	OpenNebula Measured	Eucalyptus Measured	OpenStack Measured
28,952.25	6,247.06	5,409.75	8,744.50

Stressed accompanying VM penalty

Expected Value	OpenNebula Measured	Eucalyptus Measured	OpenStack Measured
2.25	4.0	5.5	2.45

Scores:

	OpenNebula	Eucalyptus	OpenStack
bonnie++ throughput	21.58	18.69	30.20
Stressed accompanying VM penalty	56.25	40.91	100.00
μ	38.91	29.80	65.11

5.4.7 Network test

The objective of the network tests were to get an indication on the network performance under favorable conditions. Three sets of tests were executed:

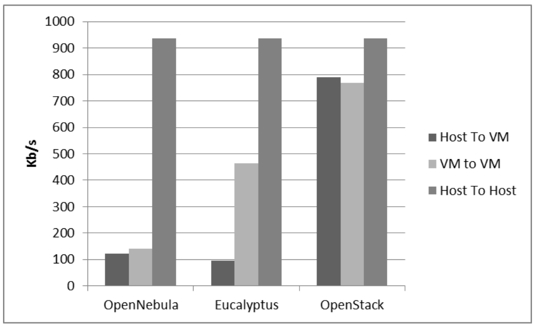
1. host-to-host to test the maximum throughput achievable on the network.
2. host-to-VM to test the best performance we can expect from a VM. It also shows the effect of the virtualization layer on the throughput.
3. VM-to-VM to test the combined effect of the virtualization layer and the network configuration.

Rank the platforms against each other by using the best throughput for Network test 3 as the expected value for the PBR scoring. The results for tests 1 and 2 will be used in Section 6.4 to calculate the the performance loss via the virtualization layer.

OpenNebula had a very low throughput if compared to the other PCMP's and we spend a few days researching the issue but could unfortunately not find a resolution. The result seems valid however as the research done by Draganov[55] also found OpenNebula to have poor network performance.

Eucalyptus had an average throughput which was a surprise as its host-to-VM performance was lower than that of OpenNebula and we thus expected a lower performance than that of OpenNebula.

OpenStack had the highest throughput with the least loss via the virtualization layer.



Network load tests

Calculations

The VM-to-VM test is used to rank the platforms, using the OpenStack throughput (kB/sec) as the expected value. The results of the host-to-host and host-to-VM tests in Section 6.4 will be used with the PCMP's overhead investigation.

Results

Expected Value	OpenNebula Measured	Eucalyptus Measured	OpenStack Measured
767	140	465	767

Scores:

	OpenNebula	Eucalyptus	OpenStack
μ	18.25	60.63	100.00

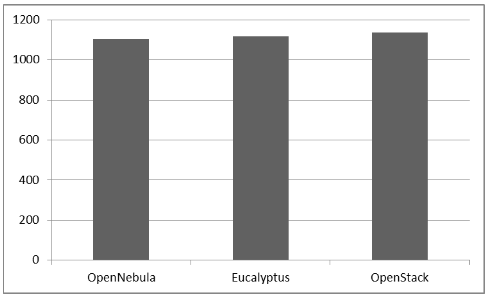
5.4.8 User applications

The objective of these tests were to get an understanding of VM performance based on real-world examples.

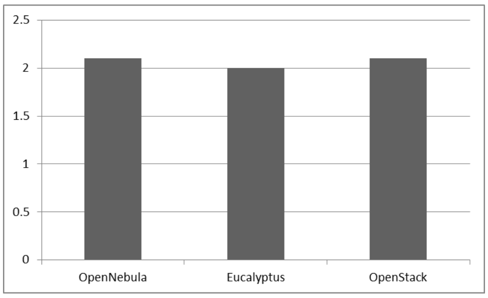
5.4.8.1 Statistical analysis

Preparation for the test consisted of copying large volumes of text log files to the Hadoop filesystem (HDFS), from outside the cloud. The test itself consisted of a RHipe analysis of the data (mostly CPU and memory intensive).

The platforms are ranked by comparing the times it take for the RHipe job to finish, using the OpenNebula value as the expected result.



Real-time (s)



VM load

The results are close to each other, as expected, as the CPU/memory requirements for the test over shadows the network and disk I/O requirements.

Calculations

Results

Expected Value	OpenNebula Measured	Eucalyptus Measured	OpenStack Measured
1105	1105	1116	1137

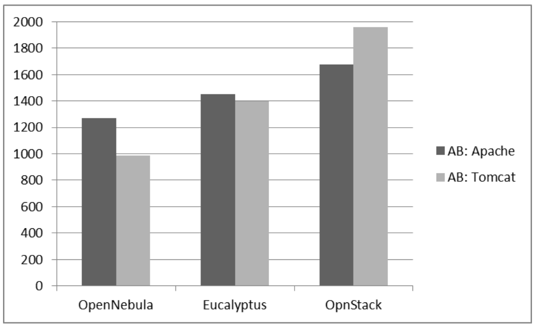
Scores:

	OpenNebula	Eucalyptus	OpenStack
μ	100.00	99.01	97.19

5.4.8.2 Web tests

The platforms are ranked by comparing the transaction rates, as well as the number of simultaneous client connections, achievable with Apache Benchmark which is executed from outside the cloud.

Note : OpenStack could only handle 200 simultaneous client connections whereas OpenNebula coped with 1000+ connections without a problem.



Web tests: Requests per second

Results

Criteria	Expected Value	OpenNebula Measured	Eucalyptus Measured	OpenStack Measured
Network Connections	1000	1000	900	200
Apache Benchmark: Apache	1673	1267	1450	1673
Apache Benchmark: Tomcat	1959	986	1400	1959

Scores:

		OpenNebula	Eucalyptus	OpenStack
Network Connections	μ	100.00	90.00	20.00
Apache Benchmark: Apache	μ	75.73	86.67	100.00
Apache Benchmark: Tomcat	μ	50.33	71.47	100.00
Average	μ	75.35	82.71	73.33
	σ	20.28	8.07	37.71

5.5 Performance results analysis

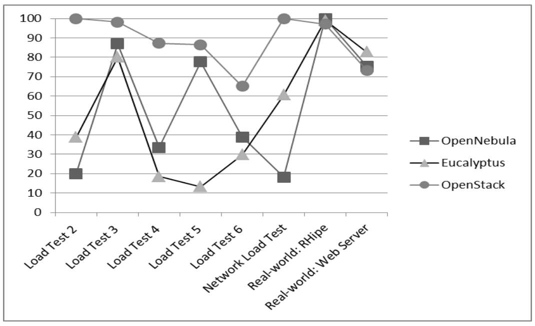
The end user's main concern, from a performance point of view, is performance penalties introduced by the PCMP and co-hosted VM's. This section uses the data collected during the load and real-world tests to score and rank the PCMP's for three use-cases, namely that of a generic system, a disk intensive system and a network intensive system. The scoring is based on the load experienced and throughput achieved by the VM's during the tests.

5.5.1 Percentage based ranking results

For the end user analysis view we deviate slightly from the system administrator calculation with the addition of the results for Load test 2, which measures the effect of a stressed host on a VM, as well as the networking related tests.

Table 5.15 shows the results used for the scoring and calculate the average across all tests to get to the final scores and standard deviations.

Table 5.15: End user load test scores



Test	OpenNebula	Eucalyptus	OpenStack
Load test 2 (bonnie++ host induced load on VM)	20.00	38.71	100.00
Load test 3 (all tests)	86.93	80.00	98.17
Load test 4 (bonnie++ throughput only)	33.55	18.48	87.17
Load test 5 (bonnie++ throughput and stressed co-VM)	77.58	13.30	86.45
Load test 6 (bonnie++ throughput and stressed co-VM)	38.91	29.80	65.11
Network load test	18.25	60.63	100.00
Real-world: RHipe	100.00	99.01	97.19
Real-world: Web server	75.35	82.71	73.33
μ	61.51	54.85	86.77
σ	28.64	31.78	12.33

OpenStack is clearly the winner for this combination of results and get not only the highest score but is also the most consistent with the lowest standard deviation.

5.6 Summary

Eucalyptus once again had various critical failures during the tests causing it to be unsuited for a production system.

OpenStack is clearly the preferred choice of the end users because of its high throughput. Its failure to handle large amounts of client network connections however raises a concern that needs to be taken into account if it is going to be used for a networking application.

OpenNebula had good scores throughout the tests, but its low network throughput is a concern.

Chapter 6

Results Analysis

6.1 Introduction

Multiple techniques to score the platforms were evaluated in order to investigate whether it is necessary to use all the KPI's available in the PCEF for an evaluation, or whether a subset would be sufficient.

The investigation covered evaluations that ranged from the very basic, using only the quantitative performance related results, all the way to the other extreme, which include all of the KPI's available.

The rest of the chapter describes the results of the different evaluation methods with suggestions in the summary on which methods to use.

6.2 PCEF scores

There might be situations where it is not practical to use the full set of PCEF KPI's for an evaluation. For example, in our evaluation we used open source platforms with similar architectures and support models. To use the complete set of KPI's would thus not make sense as it will contain a multitude of entries which will have the same score, resulting in a blurred view.

In the two sections below we will compare the PCEF scores for a comprehensive scoring, where we use the all of PCEF KPI's, to that of a limited set scoring, where we only use the critical KPI's identified in Section 2.5.5.1.

The purpose of this comparison is to investigate the effectiveness of such a partial evaluation.

Note: The priorities assigned below are based on our interpretation of the needs of the different viewpoints and must be adjusted for other evaluations.

6.2.1 Comprehensive view

This view took the whole of the PCEF into consideration by using all of the quantitative and qualitative KPI's available.

The management viewpoint focused on the purpose, financials, accountability and security; the system administrator viewpoint focused on the performance, administration, agility and accountability; and the end user's viewpoint focused on the usability, security and privacy, performance and service assurance of the system.

The following tables summarize the results:

Management viewpoint

Category	OpenNebula	Eucalyptus	OpenStack	Priority
Accountability	7.70	7.58	7.73	0.100
Agility	10.00	10.00	10.00	0.100
Assurance of Service	16.00	12.00	14.00	0.200
Finance	20.00	20.00	20.00	0.200
Performance	8.00	8.00	8.00	0.100
Security and Privacy	18.00	10.00	14.00	0.200
Usability	9.00	7.00	8.00	0.100
Score (%)	88.70	74.58	81.70	

Rank: OpenNebula $\succ$ OpenStack $\succ$ Eucalyptus

System administrator viewpoint

Category	OpenNebula	Eucalyptus	OpenStack	Priority
Accountability	8.50	8.38	8.54	0.100
Agility	7.55	6.70	8.25	0.100
Assurance of Service	39.00	12.00	30.00	0.400
Performance	22.29	21.65	17.87	0.250
Security and Privacy	4.00	2.50	3.50	0.050
Usability	9.00	5.25	7.00	0.100
Score	90.34	56.48	75.12	

Rank : OpenNebula $\succ$ OpenStack $\succ$ Eucalyptus

End user viewpoint

Category	OpenNebula	Eucalyptus	OpenStack	Priority
Assurance of Service	17.98	11.63	16.66	0.200
Performance	50.50	42.10	50.59	0.600
Usability	13.95	7.42	10.25	0.200
Score	82.42	61.13	77.51	

Rank : OpenNebula $\succ$ OpenStack $\succ$ Eucalyptus

PCEF Score The following table show the end result if all three viewpoints are taking into consideration:

Viewpoint	OpenNebula	Eucalyptus	OpenStack	Priority
Management	88.70	74.58	81.70	0.340
System administrator	90.34	56.48	75.12	0.330
End user	82.42	61.13	77.51	0.330
Score	87.15	64.06	78.11	

Rank : OpenNebula $\succ$ OpenStack $\succ$ Eucalyptus

PCEF Score, management viewpoint excluded The following table show the end result if only the system administrator and end user viewpoints are taking into consideration. We need this view to compare the results with that of the following section. The rest of the criteria was exactly the same as that of the previous section.

viewpoint	OpenNebula	Eucalyptus	OpenStack	Priority
System administrator	90.34	56.48	75.12	0.500
End user	82.42	61.13	77.51	0.500
Score	86.38	58.81	76.32	

Rank : OpenNebula $\succ$ OpenStack $\succ$ Eucalyptus

The detailed evaluation method give a consistent ranking order with all viewpoints preferring OpenNebula. The scores are also in line with the actual experience during the experiments.

6.2.2 Critical KPI view

For this view only the critical KPI's that exposed differences between platforms (as identified in Section 2.5.5.1) were taken into account. The management viewpoint was excluded as no management KPI's, that exposed severe differences between platforms, were identified during the experiments.

The following tables summarize the results.

System administrator viewpoint

Category	OpenNebula	Eucalyptus	OpenStack	Priority
Agility	8.00	7.00	7.00	0.100
Assurance of Service	25.00	16.00	21.00	0.400
Performance	28.84	26.99	15.97	0.350
Security and Privacy	4.00	2.00	3.00	0.050
Usability	6.88	3.88	5.88	0.100
Score	72.72	55.87	52.84	

Rank : OpenNebula $\succ$ Eucalyptus $\succ$ OpenStack

End user viewpoint

Category	OpenNebula	Eucalyptus	OpenStack	Priority
Assurance of Service	16.40	5.60	10.00	0.200
Performance	38.23	29.56	45.51	0.600
Usability	9.08	4.00	6.14	0.200
Score	63.71	39.16	61.65	

Rank : OpenNebula $\succ$ OpenStack $\succ$ Eucalyptus

PCEF score

Viewpoint	OpenNebula	Eucalyptus	OpenStack	Priority
System administrator	72.72	55.87	52.84	0.500
End user	63.71	39.16	61.65	0.500
Score	68.22	47.52	57.25	

Rank : OpenNebula $\succ$ OpenStack $\succ$ Eucalyptus

Once again we get a consistent ranking order with all viewpoints preferring OpenNebula.

6.2.3 Summary

The relationship of the comprehensive evaluation technique to that of the partial one can by determined by calculating the ratio between the two methods (Comprehensive Score / Partial Score) for each of the platforms evaluated:

Viewpoint	OpenNebula	Eucalyptus	OpenStack
Comprehensive Score	86.38	58.81	76.32
Partial Score	68.22	47.52	57.25
Relationship	1.27	1.24	1.33

The average difference between the two methods is 28% with a standard deviation of 3.7%. We feel this deviation is acceptable and that the partial evaluation would have been sufficient for this evaluation.

6.3 Performance based ranking

We were also interested in what the results would look like if all of the qualitative KPI's were ignored.

For this ranking we used the percentage based ranking results from Sections 4.6.1 and 5.5.1, with equal weights assigned to the system administrator and end user viewpoints.

Method	Unit	OpenNebula	Eucalyptus	OpenStack
System administrator viewpoint	μ	82.40	77.12	45.62
End user viewpoint	μ	61.51	54.85	86.77
Score	μ	71.96	65.99	66.20

Rank : OpenNebula $\succ$ OpenStack $\succ$ Eucalyptus

OpenStack is the most affected by this scoring method because of its behavior on the hosting machines. An interesting observation however is the close scores between OpenStack and Eucalyptus. This does not resemble the overall test experiences and hide some of the critical failures observed. We thus strongly advise against this, or any other performance only, scoring method.

6.4 PCMP overhead

To get an understanding of the overhead introduced by the PCMP's, we used the end user results of Load test 3 (highest VM performance test, Section 5.4.3.7) and the network load test, and scored them against the results of the bare-metal tests. The bare-metal test result is used as the expected value.

This technique is not used to score PCMP's as we accept that there will be an PCMP overhead. It can however be useful to system administrators that want to optimize their virtual environments.

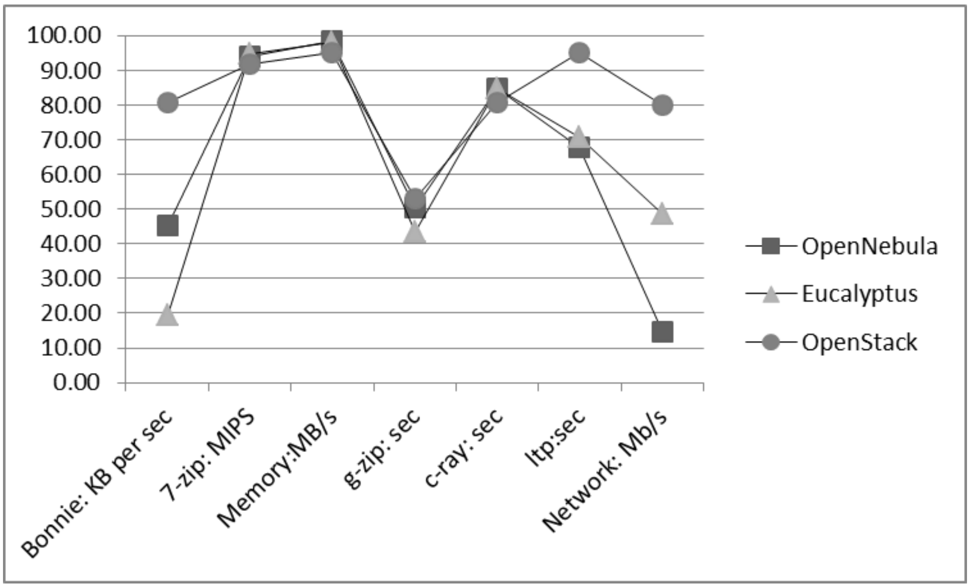


Figure 6.1: PCMP Overhead

Test	Unit	Expected Values	Observed Values OpenNebula	Eucalyptus	OpenStack
bonnie++	kB/sec	35827	16256	6923	28952
7-zip	MIPS	3750	3523	3549	3437
memory	MB/sec	3818	3750	3748	3633
g-zip	sec	17	34	39	32
C-Ray	sec	166	196	196	205
LTP	sec	1128	1665	1594	1185
Network	Mb/sec	960	140	465	767
			Normalized Values		
			OpenNebula	Eucalyptus	OpenStack

bonnie++	45.37	19.32	80.81
7-zip	93.96	94.64	91.66
memory	98.23	98.17	95.14
g-zip	50.49	43.59	53.13
C-Ray	84.84	84.69	80.85
LTP	67.70	70.72	95.19
Network	14.58	48.44	79.90
μ	65.02	65.65	82.38
σ	27.98	27.30	13.52

The result of this scoring clearly shows severe disk and network I/O penalties, which is not unexpected as it is present in the previous results as well.

The g-zip result however was unexpected because of the low score as well as the close grouping. The g-zip results will be affected by the CPU, memory and disk performance (it creates and delete a 2 GB file for each test), so we need to investigate whether the bonnie++, memory or C-Ray results can be used to explain the result. The C-Ray result partially explains this as it shows a corresponding CPU penalty for all the platforms. The 7-zip experiment used approximately two thirds the CPU resources than that of g-zip, so we would have expected the 7-zip results to be lower if it was the CPU penalty alone that was responsible for the low score. The memory penalties are negligible, leaving us with the disk I/O penalty as the main contributing factor.

We can however not explain the result if we use the bonnie++ result alone (the results range from very low to high), and we investigated the disk access differences between the bonnie++ and g-zip tests. g-zip obviously perform only a small subset of the disk operations than that of bonnie++, which explains why we don't see the same grouping pattern than that of bonnie++. The only explanation for the performance hit is the difference in file sizes; bonnie++ uses small files (kilo bytes) compared to the 2GB file of g-zip. The LTP results also hints toward the file size because a subset of its tests grows and shrinks large files.

The other interesting observation is the big difference between the above g-zip score and that of Load test 3, Section 5.4.3.7. The reason for this is the choice of the expected values used in the calculations; bare metal vs. best result. This is a good example on how sensitive the PBR is with regard to the selection of the expected value.

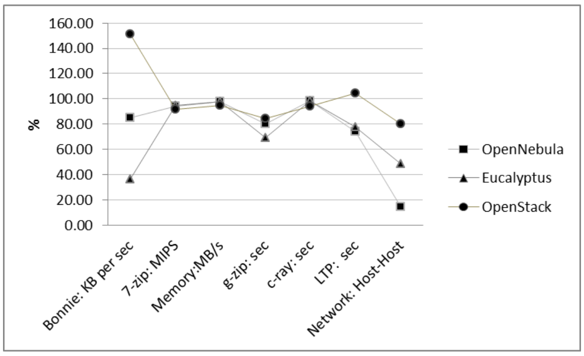
6.5 Amazon EC2 comparison

This section give an overview of how the platforms tested compares to Amazon EC2. The Amazon VM used was similar to our large VM configuration, see Table 2.3, and had the following specifications:

Table 6.2: Hardware to VM comparison

	Dual-core CPU GHz	memory (GB)	Disk (GB)	Network (Mbs)
Virtual Machine (large)	2.59	2	40	1000
Amazon EC2	2.33	2	8	

The scores are calculated by using the Amazon results as the expected value (100%). From the graph below it is clear that the primary differentiator the disk and network I/O throughput is.



	OpenNebula	Eucalyptus	OpenStack
μ	77.65	74.55	99.96
σ	27.25	22.99	22.08

Table 6.3: PCMP's vs. Amazon EC2

6.6 Summary

We now have 4 result sets, each requiring a different amount of effort, that can be used to determine whether the full set of KPI's available in the PCEF is necessary for an evaluation. The Amazon result set is excluded from the discussion as public clouds are out of scope for this thesis.

Method	Unit	OpenNebula	Eucalyptus	OpenStack
PCEF (Comprehensive)	Score	87.15	64.06	78.11
PCEF (Critical KPI's only)	Score	68.22	47.52	57.25
Performance ranking	μ	71.96	65.99	66.20
PCMP overhead ranking	μ	64.58	64.81	80.25

From the above it is clear that the performance based scorings are not in line with the PCEF scorings. We thus strongly advice against the usage of these methods.

On the other hand, if we use the critical KPI's identified by the test suite then we get a critical view of a platform that is in line with the results of a more detailed evaluation.

We thus conclude that it is not always necessary to use the full KPI set and that the subset can be sufficient to evaluate PCMP's.

Chapter 7

Related Work

7.1 Introduction

There were very few research studies on the evaluation of cloud platforms available at the time our research topic was proposed but this had changed during the course of 2011. This chapter gives a brief overview of some of the studies that relates to our work.

The work of the Cloud Services Measurement Initiative Consortium (CSMIC)[26] and Garg *et al.*[6] is of special interest as our initial work was re-organized to align with their efforts, especially with the work of CSMIC.

7.2 Existing efforts on the ranking of cloud services

CSMIC The main objective of the CSMIC is to develop a Service Measurement Index (SMI) which is a hierarchical framework for the organization and classification of cloud based services. It contains Categories (Agility, Performance, Security, etc.), associated Attributes (Capacity, Elasticity, Portability, etc.) and KPI's, or measures, that can be used to compare non-cloud to cloud based services. The current status of the SMI is that the first two levels of the framework, Categories and Attributes, have been defined with the third level, the KPI's, still outstanding.

We extended the CSMIC effort with the introduction of another layer above their Category layer by introducing the viewpoints of the enterprise decision makers (management), system administration teams that need to implement and support the system, as well as the end users that will be using the system. We also propose a set of KPI's that can be used to score the attributes. The KPI's consist of a generic set as well as a unique set that we identified during the evaluation of the CMPs.

SMICloud Garg *et al.*[6] proposed a framework, SMICloud, for the comparison and ranking of cloud services that is based on the CSMIC SMI discussed in the previous section. They do not strictly use the SMI names and refer, for example, to the SMI Financial Category as Cost.

They categorize the problem in the domain of multi-criteria decision-making (MCDM) where each individual parameter affects the service selection process. Their approach on solving the problem uses an AHP based ranking mechanism where they assign priorities to each item in the SMI levels, assuring the sum of the items in a level is exactly one. The scoring of the cloud platforms is based on the pairwise comparisons as prescribed by the AHP.

The PCEF scoring method is also based on the SMICloud multi-criteria decision-making approach with an AHP ranking mechanism that assign weights to interdependent metrics.

The PCEF do however deviate from the SMICloud in the following ways:

- The PCEF use the percentage based ranking mechanism, as described in Section 2.5.2, to represent the standard score for a metric within a specific QOS attribute. CloudSIM uses pairwise comparisons with Eigen Vectors to calculate the Relative Service Ranking Vector (RSRV).
- The PCEF does not support range types.
- Non-quantifiable measures are scored using the Likert scale.

Scoring comparison: PCEF vs SMICloud The PCEF was compared to the SMICloud using the same data as Garg *et al.* in their paper[6], see Figure 7.1, which compared three systems (S1, S2 and S3) against each other. The test data covered qualitative and quantitative KPI's in the SMI Categories of Accountability, Assurance, Financial, Performance and Security. The result comparison showed only minor differences between the two methods:

Test data

Top level QoS Groups (Weights)	First level Attributes (Weights)	Second Level Attributes (Weights)		Service 1 (S1)	Service 2 (S2)	Service 3 (S3)	Value Type	User Required Value
Accountability (.05)	level:0-10 (1)			4	8	4	Numeric	4
Agility (0.1)	Capacity (0.6)	CPU (0.5)	0.5	9.6	12.8	8.8	Numeric	4x1.6 GHZ
		Memory (0.3)	0.3	15	14	15	Numeric	10 GB
		Disk (0.2)	0.2	1690	2,040	630	Numeric	500 GB
	Elasticity (.4)	Time (1)	0.4	80-120	520-780	20-200	Range	60-120 sec
Assurance (0.2)	Availability (0.7)	0.7		99.95%	99.99%	100%	Numeric	99.9%
	Service Stability (0.2)	Upload Time (0.3)	0.3	13.6	15	21	Numeric	
		CPU (0.4)	0.4	17.9	16	23	Numeric	
		Memory (0.3)	0.3	7	12	5	Numeric	
	Serviceabilty (0.1)	Free Support (0.7)	0.7	0	1	1	Boolean	
		Type of Support (0.3)	0.3	24/7,Diagnostic Tools, Phone, Urgent Response	24/7,Diagnostic Tools, Phone, Urgent Response	24/7, Phone, Urgent Response	Unordered set	24/7, phone
Cost (0.3)	On-Going Cost (1)	VM Cost (0.6)	0.6	0.68	$0.96	0.96	Numeric	< 1 dollar/hour
		Data (0.2)	inbound	10	10	8	Numeric	100 GB/month
			outbound	11	15	18		200 GB/month
		Storage (0.2)	0.2	12	15	15	Numeric	1000 GB
Performance (0.3)	Service Response Time (1)	Range (0.5)	0.5	80-120	520-780	20-200	Range	60-120 sec
		Average Value (0.5)	0.5	100	600	30	Numeric	
Security (0.05)	level: 0-10 (1)			4	8	4	Numeric	4

Figure 7.1: SMICloud test data[6]

Results

	SMICloud	PCEF (%)
	(0.342, 0.270,0.387)	(87.52,72.27,99.44)
Rank	S3 $\succ$ S1 $\succ$ S2	S3 $\succ$ S1 $\succ$ S2

The ranking order for both methods is the same with S3 preferred to S1, which in turn is preferred to S2. The ratio of the scores, when measured against each other, is also similar between the two methods.

7.3 Public cloud evaluation efforts

Ostermann *et al.*[56] and Iosup *et al.*[57]'s research to answer the question 'Is the performance of clouds sufficient for MTC-based scientific computing?'. Their main finding was that the compute performance of the public clouds, including Amazon, they tested is too low to be used for scientific computing. Based on our findings as well as the findings of Pelletingeas[53] with regards to the performance of Amazon vs. the platforms we tested, we can probably conclude that the private cloud platforms will have the same issues as the public clouds, if the same low end hardware we used for our tests are used to build the private cloud.

Other The interest in cloud computing is gaining momentum and papers, such as the work of Chhetri *et al.*[58], that defines a test suite to do performance benchmarking on public clouds, are starting to appear. They mention quite a few other commercial and open source tools and frameworks that can be used for performance based public cloud benchmarking purposes. These tools and frameworks follows similar procedures as the ones defined in our test suite, and in some cases it uses the same tools we use for matching use cases.

We do however need to re-iterate our warning against the usage of these performance only based evaluations for the reasons given in Section 6.6.

7.4 Private vs. public clouds

Baun et al's, [59] research on Eucalyptus's performance compared to Amazon Baun et al did research to determine if the performance of Eucalyptus is acceptable for it to be used as a private cloud infrastructure. Their research covered performance testing on the Eucalyptus disk, CPU and network subsystems which they then compared to the results of the same tests when ran on Amazons public cloud.

They conclude that the performance of Eucalyptus is almost the same as that of Amazon and in some cases even better, but that Eucalyptus is unfortunately not as stable as Amazon .

Their performance finding is not in line with our results but this can be contributed to the hardware they used, servers compared to our workstations. This observation is key to the selection of hardware to be used for a private cloud. There is a perception that commodity hardware = cheap hardware. This is not the case and the machines selected for a private cloud should lean towards high-end servers if performance is important.

The stability finding however is in line with ours, which is a bit of a concern as our experiment was done 2 years after their effort.

Pelletingeas[53]'s research on OpenNebula's performance compared to Amazon Christophe Pelletingea's research for his M.Sc thesis included a performance study of OpenNebula compared to Amazon. His tests were similar to that of ours with the exception of his choice of hypervisor, Xen.

He concluded that the performances of private clouds, or more accurately OpenNebula, is similar to that of public clouds, Amazon in his case, and that the overhead of the hardware assisted virtualization layer is acceptable for CPU performance, but that I/O performance needs improvement.

Our results however shows that OpenNebula scored a low 77.65% if compared to Amazon and that the virtualization loss is almost 35%, but this might again be contributed to his hardware configuration. We do however agree with the CPU and I/O performance results as it is line with ours.

Aleksandar Draganov [55]'s research on Hybrid clouds using OpenNebula and Amazon Aleksandar Draganov's M.Sc project investigated the suitability of OpenNebula as a hosting platform for compute intensive web applications. His findings are in line with ours and also concluded that OpenNebula is simple to install and use, and that the network throughput is low.

7.5 Summary

This chapter summarized existing efforts on the ranking of cloud services, some of which we used as the foundation of our work.

We also pointed out that our performance based findings are mostly in line with that of the other efforts, and that the changes can probably be contributed to different hardware configurations used during the experiments.

Chapter 8

Conclusions and future work

8.1 Introduction

In the thesis introduction we explained the concept of cloud computing, more specifically private cloud computing, and motivated why we need to evaluate PCMP's in a controlled manner.

We also identified the goals for the paper which contained a definition for private cloud computing, a reference model for the evaluation of PCMP's and an implementation of the reference model. This chapter conclude the thesis by evaluating the progress made towards reaching these goals. It also provides a section that propose possible future work in the field of PCMP's, as well as enhancements to the PCEF.

8.2 Concluding statements

We start with a short discussion around the progress made towards achieving the goals set in Section 1.3

8.2.1 Private cloud computing definition

If we use the definition for private cloud computing as defined in Section 1.5 then, based on the test results, we can either conclude that the PCMP's we tested were immature, or we need to re-visit the definition. This is because none of them implemented a meaningful subset of the features required by the definition, hence their low scores for the automation and self-service requirements.

Commercial private cloud offerings such as VMWare however provide a good subset of the functionality as per the definition so it is not an unfair definition and we therefore stand by it.

8.2.2 Private Cloud Evaluation Reference Model and Framework

Our reference model is based on the concept of an evaluation framework that can score PCMP's based on the requirements of different stakeholders.

Our implementation of the model, the PCEF, proved itself to be flexible and easy to use. It allows for the evaluation of PCMP's with detail levels ranging from very basic, to very detailed if needed. This is crucial for a framework like the PCEF as the framework is too comprehensive to be practical for all evaluations.

The sections to follow give in-depth discussions of the framework's test suite and scoring mechanism as it is essential for the success of it.

8.2.2.1 Test suite and PCEF KPI's

The platforms evaluated with the test suite had similar architectures which created the expectation that the load test results would be in line with each other if the same Hypervisor and VM were used for all the platforms.

The test suite developed in this work however managed to discover various performance related differences between the platforms. We used these differences to create PCEF specific KPI's that need special consideration during the evaluation of private clouds, as they have the ability to identify shortcomings that separate the platforms from each other.

8.2.2.2 Scoring mechanism

In the private cloud definition section we pointed out that the preferred PCMP would have to compromise between the requirements of system administrators and those of the end users. The PCEF succeeded in this by ranking OpenNebula in the first place which, is in line with our experience with the platforms during the evaluations.

The PCEF also showed that we can use neither a featured based evaluation, for example the CMPs Quadrant in Figure 3.1, nor a performance based evaluation, as discussed in Section 6.6, for the selection of a PCMP as these techniques gives misleading results.

8.2.3 Summary

Taking all of the above into consideration, we conclude that the reference model and the PCEF are both practical and required for the evaluation and scoring of PCMP's.

The effectiveness of our framework was demonstrated by it exposing various strengths and weaknesses in the platforms evaluated, and we noted that the final ranks of the PCMP's evaluated a fair reflection of the actual evaluation experience is.

We therefore conclude that we successfully created a framework that can be used to easily score and rank private cloud management platforms and that we achieved the original goal set in the introduction.

8.3 Recommendations for further work

A number of shortcomings and potential improvements for PCMP's and the PCEF were identified during the compilation of the thesis:

- The current implementation of the PCEF contains manual processes that can be automated.
- The PCEF managed to identify many unexpected performance related differences between the platforms tested. We therefore feel that there is a need for an optimization framework that can be used to dynamically, without service interruption, optimize VM's based on the use cases deployed.
- We also noted a large shortcoming in the user interface and automation components of the platforms tested. Critical features like work-flow components, virtual machine image creation, cloning and snapshots, deployment configuration were either rudimentary or lacked completely.

All of the platforms tested support the Amazon Web Services (AWS) API, so it would be worthwhile to develop a generic PCMP management and diagnostic system, using the AWS API to fill this gap. The work of Liu [60] can be used as the basis for such a system.

Appendices

Appendix A

PCEF KPI Details

A.1 Introduction

Below is the complete list of KPI's identified for the PCEF. The majority is from the 'General Package Evaluation Criteria Template' published by Borysowich[48] and the 'Research Challenges for Enterprise cloud computing' paper [49].

The main sections represent the SMI Categories and the sub sections the SMI Attributes, where the KPI's are listed.

A.2 Accountability

Can we count on the software provider organization and community?

A.2.1 Auditability

The ability of a client to verify that the service provider is adhering to the standards, processes, and policies that they follow:

- SDLC Folows COBIT, TOGAF, CMMI or ISO practices and standards.
- Source code is available.

A.2.2 Compliance

Standards, processes, and policies that the service provider committed to are followed:

- A methodology was used to develop the products.
- Contract protects customer from damages due to non-performance of software.
- Documentation updated for any fixes.
- Ownership of components clearly defined.
- Ownership of modifications to source code clearly defined.
- Payments are milestone/delivery based.
- Remedies for breach of contract provided.
- Satisfactory right to terminate clause in contract.
- Source code available in hard copy and electronic media.
- Warranties for performance available.

A.2.3 Contracting experience

Indicators of client effort and satisfaction with the process of entering into the agreements required to use a service:

- Cloud Provider has standard contracts in place for Enterprise level support.

A.2.4 Data ownership

The Level of rights a client has over client data associated with a service:

- Enterprise has full ownership of data.

A.2.5 Ease of doing business

Client satisfaction with the ability to do business with a service provider:

- Active user group exists for each product.
- Location of customer service employees (in relation to customer):
  - Same city.
  - Same region.
  - Same country.
- Published evaluations of software provided.
- Sample plans provided (e.g., implementation, training).
- Support Community activity is healthy.
- Telephone service provided:

  - 24 hours per day, 7 days per week.
  - 24 hours per day, 5 days per week.
  - 8 hours per day, 7 days per week.
  - 8 hours per day, 5 days per week.
- User Group influences release of the product
- Willing to demonstrate products:
  - At customer site.
  - At vendor site.

A.2.6 Governance

The processes used by the service provider to manage client expectations, issues and service performance:

- Guaranteed response time (e.g., customer representative responds with either fix or action plan) within:
  - 1 to 2 hours.
  - 3 to 5 hours.
  - 6 to 12 hours.
  - 24 hours.
- Maximum time to implement an emergency fix:
  - 1 to 2 days.
  - 3 to 5 days.
  - 6 to 10 days.
  - 11+ days.
- Procedures for vendor-initiated fixes provided.
- Satisfactory policy on supporting past releases.

A.2.7 Ownership

The level of rights a client has over software licenses, intellectual property and data associated with a service:

- Customer can modify software without impacting warranty or support.

A.2.8 Provider business stability

The likelihood that the service provider will continue to exist throughout the contracted term:

- Acid test ratio ((Cash + Marketable Securities + Current Receivables) / Current Liabilities) is 0.75:1 or more.
- Age of product (in years):
  - 0 to 2.
  - 3 to 5.
  - 5+.
- Consistent financial position over last three years:
  - very consistent.
  - somewhat consistent.
  - inconsistent.
- Equity / Assets is at least 30
- Equity / Debt is at least 50
- Gross Margin on Sales (Net Sales Revenue - Cost of Goods Sold) / Net sales Revenue:
  - 0 to 30.
  - 31 to 40.
  - 41 to 50.
  - 51+.
- No significant cautions in Dun & Bradstreet report.
- No significant cautions in financial statement notes (e.g., legal actions, tax assessments, contingent losses or gains, large lump sums due).
- Number of customers (e.g., companies, organizations) using each product:
  - 0.
  - 1 to 9.
  - 10 to 24.
  - 25+.
- Number of employees:
  - 1 to 100.
  - 101 to 500.
  - 501 to 1000.
  - 1001+.
- Number of new customers in last 24 months (as a percentage of customer base):
  - 0 to 5.

  - 6 to 10.
  - 11 to 25.
  - 26+.
- Number of releases (functional enhancements) for each product in last 5 years:
  - 0 to 1.
  - 2 to 3.
  - 4 to 5.
  - 6+.
- Number of these releases still being supported:
  - 0 to 1.
  - 2 to 3.
  - 4 to 5.
  - 6+.
- Percentage of sales revenue allocated for R&D on each product:
  - 0 to 1.
  - 2 to 5.
  - 6+.
- Proof of success in similar organization provided.
- Reference sites provided for each product.
- Return (before taxes) on Shareholders' Investment ((Net Income before extraordinary items - Preferred Dividends) / (Average Total Shareholders Equity - Preferred Shares)):
  - 0 to 5.
  - 6 to 10.
  - 11 to 15.
  - 16+.
- Return (before taxes) percent on sales (Income before extraordinary items and taxes / Net Sales):
  - 0 to 5.
  - 6 to 10.
  - 11 to 15.
  - 16+.
- Times Interest Earned (Earnings before interest expense and income taxes / Interest Expense) is at least 1.5.

- Working Capital Ratio (Current Assets / Current Liabilities) is 1:1 or more.
- Years in business:
  - 0 to 2.
  - 3 to 5.
  - 5+.

A.2.9 Provider contract/Service Level Agreement (SLA) verification

The service provider makes available to clients SLAs adequate to manage the service and mitigate risks of service failure:

- Failure to install an update, enhancement, or new release impacts the warranty/support/maintenance after:
  - 30 days or less.
  - 31 to 60 days.
  - 61 to 90 days.
  - 91+ days.
- Proposed contract provided.
- Software license agreements provided (e.g., software maintenance, support).
- Software warranty begins at:
  - Shipment.
  - Receipt.
  - Installation.
  - Acceptance.
- Updates, enhancements and new releases covered under maintenance agreements.
- Warranty/support/maintenance continues after a product function or the entire product is discontinued for:
  - 30 days.
  - 31 to 60 days.
  - 61 to 120 days.
  - 120+ days.
- Warranty/support/maintenance is provided for modifications specifically

requested by the customer.

A.2.10 Provider personnel requirements

The extent to which service provider personnel have the skills, experience, education, and certifications required to effectively deliver a service:

- Consulting Services Focus on best practices in the successful design of cloud architectures.
- Number of developers of products that are currently employed by vendor (i.e., original developers are available for enhancements):
  - 1 to 10.
  - 11 to 25.
  - 25+.
- Employees are assigned to product development for each product in sufficient numbers to allow for regular maintenance and enhancements.
- Engineering services assist IT staff through each stage of deployment and integration to help them install, configure and operate the cloud system.
- Evaluation support program.
- Support services to assess its suitability and performance in your environment.
- Number of employees in customer service for each product per installed user base is sufficient to meet service levels.
- On-site expertise available at no or low cost.
- Technical support subscription (Provides long term multi-year support, production-level support with professional SLAs, guaranteed patches, regular updates and upgrades, product influence, and privacy and security guarantee).
- Training (Train IT staff with the skills they need to install, configure, customize and operate the cloud platform).
- Vendor's customer service employees are available for implementation.

A.3 Agility

Can it be changed, and how quickly can it be changed?

A.3.1 Adaptability

The ability of the service provider to adjust to changes in client requirements:

- Customization and developed interfaces will not impact upgrades to product.
- System is upgradeable without migration to another platform.

A.3.2 Elasticity

The ability of a service to adjust its resource consumption to meet demand:

- Agility (How quickly the provider responds as the consumer's resource).
- Elasticity. The ability for a given resource to grow, with limits (the maximum amount of storage or bandwidth, for example) clearly stated.
- Incremental growth path to permit expansion, as needed.

A.3.3 Extensibility

The ability to add new features or services to existing services:

- Programming languages supported:
  - C.
  - C++.
  - Java.
  - Python.
  - Ruby.
  - etc.

A.3.4 Portability

The ability of a client to easily move a service from one service provider to another with minimal disruption:

- Minimal customization to products required.

A.3.5 Scalability

The ability of a service provider to increase or decrease the amount of service available to meet client requirements:

- The ability to service a theoretical number of users.

A.4 Assurance of Service

How likely is it that the service will work as expected?

A.4.1 Availability

Availability is determined by the internal deployment architecture. It represent the amount of time that a client can make use of a service.

Availability	Downtime per year	Downtime per month	Downtime per day
99.99999% (seven nines)	3.15 seconds	0.259 seconds	0.0605 seconds
99.999% (five nines)	5.26 minutes	25.90 seconds	6.05 seconds
99.99% (four nines)	52.56 minutes	4.32 minutes	1.01 minutes
99.9% (three nines)	8.76 hours	43.80 minutes	10.10 minutes

A.4.2 Recoverability

Recoverability is the degree to which a service is able to quickly resume a normal state of operation after an unplanned disruption:

- Data is recoverable after computer failure with minimal operator assistance, within 24 hours.
- Recovery mechanism ensures data integrity to the business function level.

A.4.3 Reliability

Reliability reflects measures of how a service operates without failure under given conditions during a given time period:

- DNS / DHCP Services function properly.
- Durability (How likely the data is to be lost).
- GUI is provided and functional.
- Reliability (How often the service is available).
- The overhead of the cloud controller on the system as a whole is acceptable.
- VM is healthy after start-up.

A.5 Finance

How much does it cost?

A.5.1 Acquisition and transition cost

Any client costs to acquire the rights and ability to use a service and to move from an existing service to the new one:

- All documentation included in price.
- Conversion support costs identified.
- Copies of documentation included in price:
- Customization costs identified.
- Implementation support costs identified.
- No or low costs for additional copies of documentation.
- One-time costs are included in pricing.
- Prices of other products that must be licensed/purchased for use with each product (e.g. Database license) identified.
- Pricing arrangements available (purchase, lease)
- Training costs identified.
- Training days included in price.
- Training materials include in price.

A.5.2 On-going cost

The client cost to operate a service. This includes both recurring flat costs (e.g., monthly access fees) and usage-based costs:

- Incremental costs for upgrades and expansions are included in pricing.
- Invoices will be payable in appropriate currency.
- No or low price increases (including maintenance/support) over last 3 years.
- On-line help facilities included in price.
- Pricing structure for follow-up training provided.
- Recurring annual and monthly costs are included in pricing.

A.6 Performance

Does it do what we need?

A.6.1 Functionality

The specific features provided by a service:

- Audit trails automatically provided for all transaction detail.
- Hypervisor support:
  - Hyper-V (Microsoft).
  - KVM.
  - LXC.
  - Oracle VM.
  - QEMU.
  - VMWare ESX/ESXi.
  - Xen.
- Reporting tools are provided for browsing audit trails.
- System capable of enabling before and after logging feature.

A.6.2 Interoparabillity

The ability of a service to easily interact with other services (from the same service provider and from other service providers):

- Amazon Web Services.
- EC2 Query.
- Hybrid cloud computing and cloud bursting.
- Open Cloud Computing Interface (OCCI)13.
- Other software products that could be integrated with these products are available.
- Portability across cloud vendors.
- vCLoud.

A.6.3 Service response time

An indicator of the time between when a service is requested and when the response is available:

- Customer service response times. How quickly the provider responds to a service request. This refers to the human interactions required when something goes wrong with the on-demand, self-service aspects of the cloud.

A.6.4 Suitability

How closely do the capabilities of the service match the needs of the client:

- Linearity How a system performs as the load increases.
- Percentage based ranking scores for all load tests.

A.7 Security and Privacy

Is the service safe and is privacy protected?

A.7.1 Access control and privilege management

Policies and processes in use by the service provider to ensure that only the provider and client personnel with appropriate status/reasons to make use of or modify data/work products may do so:

- Access control of data (read, write, delete, copy) at the file level by user group provided.
- Automatic password expiry definable.
- Data attribute (field) level security provided.
- File level security by user group provided.
- Functional security by user group provided.
- Name and password control provided.
- Passwords never display on monitors.
- Provides central control of passwords.
- Security violations are automatically logged.
- Stored passwords are encrypted.

A.7.2 Data geographical/political

The client's constraints on service location based on geography or politics:

- Distributed geographical deployments possible.

A.7.3 Data privacy and data loss

Client restrictions on access and use of client data are enforced by the service provider. Any failures of these protections are promptly detected and reported to the client:

- Data Loss and Reliability.
- Data Protection and Confidentiality.
- Logical multi-tenancy to manage VM attacks, network sniffing and malicious code execution.
- Secure APIs to manage:
  - Man-in-the-middle attacks.
  - Content threats.
  - Code injection.
  - Disk security.

A.7.4 Proactive threat and vulnerability management

Mechanisms in place to ensure that the service is protected against know recurring threats as well as new evolving vulnerabilities:

- Account hacking, access control, and authorization. Coarse account access control at the cloud provider increases the value of a stolen account.
- Cloud provider insider (malicious insider) threats. Mismatched security practices at the cloud service provider creates a weak link for a determined attacker.
- Intrusion prevention system available.

A.7.5 Retention and disposition

The service provider's data retention and disposition processes meet the clients' requirements:

- The service provider's data retention and disposition processes meet the clients' requirements.

A.8 Usability

Is it easy to learn and to use?

A.8.1 Installability

Installability characterizes the time and effort required to get a service ready for delivery (where applicable):

- Installation is straight forward and repeatable.
- VM access configuration is straight forward.

A.8.2 Learnability

The effort required of users to learn to use the service:

- Classroom instruction and hands-on training provided.
- Computer based training software provided.
- Multi-language training and training materials provided, as required.
- System administration training provided.
- Training in functional use provided.
- Training pre-requisites defined; e.g. basic keyboarding skills.
- Training program can be customized to meet specific customer requirements.
- Training provided at:
  - Customer site(s).
  - Vendor site(s).

A.8.3 Operability

The ability of a service to be easily operated by users:

- Consistent user Interface; e.g., screen layouts, keyboard functions, and navigation.
- Context sensitive help provided (e.g., F1 from any window field).
- Descriptions of error messages are understandable.
- GUI interface (e.g., windows, icons, mouse, pull-down menus) consistent with CUA or similar standard.
- Layered architecture with maximum independence between layers.
- Makes consistent, effective use of colour.
- Menus can be customized.
- Menus provide access to all applications.
- Minimal effort required to escape from incorrect selection or system error.
- Minimum two levels of error messages provided: error and warning.

- Numeric entries automatically signed where possible (e.g., credit displays as negative).
- On-line help provided.
- Features to assist in locating a function or feature.
- Descriptions of how each function works.
- Descriptions of fields, their contents, and acceptable formats.
- Operates on commodity hardware.
- Platform independent.
- Provides formatting options for currency (e.g., amounts formatted with commas or spaces).
- Provides formatting options for dates (e.g., DD/MM/YY, MM/DD/YY, YY/MM/DD) that can be easily changed from one format to another.
- Provides on-line look-ups of key field information (e.g., customer by ID or name).
- Provides on-line look-ups on partial keys (e.g., part of customer name).
- Provides single system image.
- Quick paths provided from one function to another without using menus.
- Single point of entry provided for all data input.
- The system administration capabilities of the Cloud System is adequate for a production environment
- Tools provided to create custom reports.

A.8.4 Suitability

How closely do the capabilities of the service match the needs of the client:

- Acceptable resource requirements (e.g., memory, storage, I/O channels, special) defined and costed.
- Accessible from remote locations.
- Backup and recovery.
- Common system administration operations is automated.
- Cost management and allocation.
- Data are shared in same file structures across sub-systems.
- Data is replicated to at least 2 other systems.
- Designed for easy installation and support.
- Designed for use in a distributed environment.
- Disaster recovery and business continuity.

- Enhanced monitoring, alerts and escalation mechanism provided.
- Inventory Management.
- Load balancing.
- Localized storage for MapReduce types of applications and temporary data.
- OS and software patch management.
- Pre-configured components available.
- Self-healing.
- Self-service portals for infrastructure and platforms.
- Software license management and reporting.
- Sufficient auditing and logging.
- Support for enterprise Linux.
- Support for Microsoft Windows
- The system administration capabilities of the cloud platform is adequate for a production environment.
- Visibility in resource costs and usage.
- VM (de)commissioning is easy via a UI.
- VM configuration of firewall for VM's.
- VM full control of VM instance life-cycle via a user interface.
- VM hook manager to trigger administration scripts upon VM state change.
- VM image repository subsystem with catalog.
- VM online backups possible.
- VM over-subscribing.
- VM snapshots possible.
- VM template repository subsystem.

A.8.5 Understandability

The ease with which users can understand the capabilities and operation of the service:

- Documentation can be copied by customer without restriction.
- Documentation is available in electronic format (e.g., CD-ROM).
- Documentation is regularly updated and distributed.
- Documentation may be maintained by the customer.
- Installation guide available.

- Maintenance guide available.
- Operating documentation available.
- System documentation available.
- System documentation includes design and setup information.
- Training documentation available.
- User and training documentation describes how to use each component in non-technical terms (e.g., functional description).
- User documentation available.
- User documentation clearly defines procedures for all processes.

List of References

[1] Gartner, Inc.: Gartner identifies the top 10 strategic technologies for 2010, 2009. Available at: http://www.gartner.com/it/page.jsp?id=1210613

[2] Gartner, Inc.: Gartner identifies the top 10 strategic technologies for 2011, 2010. Available at: http://www.gartner.com/it/page.jsp?id=1454221

[3] Mell, P. and Grance, T.: The nist definition of cloud computing. Report 800-145, National Institute of Standards and Technology, 2011.

[4] DZone: DZone's Definitive Guide to PaaS & IaaS Solutions. Tech. Rep., DZone, 2013.

[5] Llorente, I.M.: Eucalyptus, cloudstack, openstack and opennebula: A tale of two cloud models, 2013.
Available at: http://blog.opennebula.org/?p=4042

[6] Garg, S.K., Versteeg, S. and Buyya, R.: SMICloud: A Framework for Comparing and Ranking Cloud Services. In: *IEEE International Conference on Utility and Cloud Computing*. 2011.

[7] CSMIC: Service measurement index, version 1.0. Tech. Rep., Carnegie Mellon University Silicon Valley, 2012.

[8] Wikipedia: 2013.
Available at: http://en.wikipedia.org/wiki/Analytic_hierarchy_process

[9] Amazon: Amazon web services, 2013.
Available at: http://aws.amazon.com/

[10] Eucalyptus Systems, Inc.: Eucalyptus: open source aws compatible private clouds, 2012.
Available at: http://www.eucalyptus.com/

[11] RedHat: Kvm, 2011.
Available at: http://www.linux-kvm.org/page/Main_Page

[12] Carnegie Mellon University: Carnegie mellon university, silicon valley, 2012.
Available at: http://www.cmu.edu/silicon-valley/

[13] Cambridge University Press: Cambridge dictionaries online, 2013.
Available at: http://dictionary.cambridge.org/dictionary/british/framework?q=framework

[14] Apache: Hadoop, 2011.
Available at: http://hadoop.apache.org/

[15] Open Grid Forum: Open cloud computing interface, 2012.
Available at: http://occi-wg.org/

[16] Bates, D., Chambers, J., Dalgaard, P. *et al.*: The r project for statistical computing, 2011.
Available at: http://www.r-project.org/

[17] ubuntu.com: Ubuntu cloud infrastructure, 2012.
Available at: https://help.ubuntu.com/community/UbuntuCloudInfrastructure

[18] Gartner Inc: Gartner says nearly half of large enterprises will have hybrid cloud deployments by the end of 2017, 2013.
Available at: http://www.gartner.com/newsroom/id/2599315

[19] VMWare: Cloud operations:a new model for service delivery, 2013.
Available at: http://www.vmware.com/files/pdf/vmware-cloudops-servicedeliverymodels-white-paper.pdf

[20] Unisys: The unisys hybrid enterprise, 2013.
Available at: http://www.unisys.com/unisys/ri/pub/bl/detail.jsp?id=1120000970015710154

[21] OpenNebula.org: The open source solution for data centre virtualization, 2012.
Available at: http://opennebula.org

[22] OpenStack: Openstack open source cloud computing software, 2012.
Available at: http://www.openstack.org

[23] Amazon: Amazon elastic compute cloud (amazon ec2), 2012.
Available at: http://aws.amazon.com/ec2/

[24] VMWare: Controlling virtual machine sprawl, 2012.
Available at: http://www.vmware.com/files/pdf/techpaper/VMware-Controlling-Virtual-Machine-Sprawl.pdf

[25] VMWare: vCloud Suite: Infrastructure-as-a-Service (IaaS) & Cloud, 2013.
Available at: http://www.vmware.com/products/vcloud-suite/

[26] Cloud Services Measurement Initiative Consortium (CSMIC): Service mesurement index (smi), 2012.
Available at: http://csmic.org/about-csmic/

[27] Blogger: Treffis: Intel's x86 servers will dominate but ARM's entry will take share, 2013.
Available at: http://www.trefis.com/stock/intc/articles/192045/intels-x86-servers-will-dominate-but-arms-entry-will-take-share/2013-06-18

[28] VentureBeat: ARM-based OpenStack: it's like an army of cell phones powering cload-based computing, 2013.
Available at: http://venturebeat.com/2012/07/18/arm-based-openstack-its-like-an-army-of-cell-phones-powering-cloud-based-computing/

[29] Leverhawk: What does ARM mean for the future of the cloud?, 2013.
Available at: http://leverhawk.com/what-does-arm-mean-for-the-future-of-the-cloud-20121219119

[30] Date Center Knowledge: Baidu deplyos Marvell ARM-based cloud server, 2013.
Available at: http://www.datacenterknowledge.com/archives/2013/02/28/baidu-deploys-marvell-arm-based-server/

[31] TryStack: 2013.
Available at: http://www.trystack.org/

[32] Guha, S.: Rhipe documentation, 2011.
Available at: http://www.rhipe.org

[33] Kabacoff, R.I.: R in action, 2011.

[34] Lam, C.: Hadoop in action, 2011.

[35] The Apache foundation: The apache http server project, 2013.
Available at: http://httpd.apache.org/

[36] The Apache foundation: Apache tomcat, 2013.
Available at: http://tomcat.apache.org/

[37] The Apache foundation: ab - apache http server benchmarking tool, 2013.
Available at: http://httpd.apache.org/docs/2.4/programs/ab.html

[38] SearchCloudComputing: Five open source tools for building and managing clouds, 2012.
Available at: http://searchcloudcomputing.techtarget.com/tip/Five-open-source-tools-for-building-and-managing-clouds

[39] ubuntu.com: 2011.
Available at: https://help.ubuntu.com/community/UEC/Topologies

[40] Linux Test Project: Linux test project, 2012.
Available at: http://ltp.sourceforge.net/

[41] Coker, R.: bonnie++, 2001.
Available at: http://www.coker.com.au/bonnie++/

[42] Jones, R.: netperf, 2011.
Available at: http://www.netperf.org

[43] Phoronix.com: phoronix-test-suite, 2012.
Available at: http://www.phoronix-test-suite.com/

[44] Massie, M.: 2011.
Available at: http://ganglia.sourceforge.net/

[45] nixCraft: 20 linux system monitoring tools every sysadmin should know, 2013.
Available at: http://www.cyberciti.biz/tips/top-linux-monitoring-tools.html

[46] Skopal, J.: jnettop wiki, 2013.
Available at: http://jnettop.kubs.info/wiki/

[47] Collectl: Collectl, 2012.
Available at: http://collectl.sourceforge.net/

[48] Borysowich, C.: Software evaluation criteria templates: General package evaluation criteria template, 2006.
Available at: http://it.toolbox.com/blogs/enterprise-solutions/software-evaluation-criteria-templates-13455

[49] Khajeh-Hosseini, Sommerville and Sriram: Research challenges for enterprise cloud computing. Tech. Rep., University of St Andrews, Cloud Computing Co-laboratory, School of Computer Science, 2013.

[50] Younge, A.J., Henschel, R., Brown, J.T., von Laszewski, G., Qiu, J. and Fox, G.C.: Analysis of Virtualization Technologies for High Performance Computing Environments. In: *IEEE International Conference on Cloud Computing*, pp. 9–16. 2011.

[51] Intel: Hardware-assisted virtualization technology, 2013.
Available at: http://www.intel.com/content/www/us/en/virtualization/virtualization-technology/~hardware-assist-virtualization-technology.html

[52] AMD: Amd virtualization, 2013.
Available at: http://www.amd.com/us/solutions/servers/virtualization/Pages/virtualization.aspx

[53] Pelletingeas, C.: *Performance Evaluation of Virtualization with Cloud Computing*. MSc Advanced Networking, Edinburgh Napier University, School of Computing, 2010.

[54] Red Hat, Inc.: Overcommitting resources, 2013.
Available at: https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/5/html/Virtualization/sect-Virtualization-Tips_and_tricks-Overcommitting_with_KVM.html

[55] Draganov, A.: *Exploiting Private and Hybrid Clouds for Compute Intensive Web Applications*. MSc in High Performance Computing, The University of Edinburgh, 2011.

[56] Ostermann, S., Iosup, A., Yigitbasi, N., Prodan, R., Fahringer, T. and Epema, D.: An early performance analysis of cloud computing services for scientific computing. Tech. Rep. PDS-2008-006, Delft University of Technology, 2008.

[57] Iosup, A., Ostermann, S., Yigitbasi, N., Prodan, R., Fahringer, T. and Epema, D.: Performance analysis of cloud computing services for many-tasks scientific computing. Tech. Rep., IEEE, 2010.

[58] Chhetri, M.B., Chichin, S., Vo, Q.B., and Kowalczyk, R.: Smart cloud broker - finding your home in the clouds. Tech. Rep., Swinburrne University of Technology, Melbourne, Victoria 3122, Australia, 2013.

[59] Baun, C. and Kunze, M.: Building a private cloud with Eucalyptus. In: *IEEE International Conference on E-Science Workshops*. 2009.

[60] Liu, C.: *Automated cloud resource orchestration*. Degree of Doctor of Philosophy, University of Pennsylvania, 2012.