

A Three-Dimensional Performance Measurement Model for Cloud Computing

Luis Eduardo Bautista Villalpando¹, Alain Abran², Anderson Ravello², Arturo Elías Ramírez¹

¹Department of Electronic Systems, Autonomous University of Aguascalientes, Aguascalientes, Mexico

²Department of Software Engineering and Information Technology, ETS—University of Quebec, Montreal, Canada

Email: lebautis@correo.uaa.mx, alain.abran@etsmtl.ca, ravello@gmail.com, aeliasr@correo.uaa.mx

How to cite this paper: Villalpando, L.E.B., Abran, A., Ravello, A. and Ramírez, A.E. (2018) A Three-Dimensional Performance Measurement Model for Cloud Computing. *Journal of Software Engineering and Applications*, 11, 235-248. <https://doi.org/10.4236/jsea.2018.115015>

Received: April 29, 2018

Accepted: May 28, 2018

Published: May 31, 2018

Copyright © 2018 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

New paradigms for processing and storing data such as cloud computing require new approaches for the measurement of cloud service performance. To establish a Service Level Agreement (SLA) between a cloud service provider and its customers, the cloud services and their service level objectives need to be identified. An additional challenge in the performance measurement of cloud services is the lack of models that integrate the different perspectives of providers, maintainers and customers within the same model in order to define the concepts commonly used in cloud SLA contracts. This work proposes a three-dimensional Performance Measurement Model for Cloud Computing (P2M2C-3D) which consolidates performance measurement from the perspectives of providers, maintainers and customers for the different types of cloud services.

Keywords

Cloud Computing, Performance, Measurement, Model, Services, Maintenance, QoS, SLA

1. Introduction

In order to design a measurement process and its measurement model, one must define the activities that are required to adequately specify what measurement information is required, how the measurement and analysis results are to be applied, and how to determine if the analysis results are valid according to ISO/IEC 15939:2008-Systems and Software Engineering-Measurement process [1]. Measuring the performance of computer systems is important because it makes it possible to determine whether a system meets the requirements under stated conditions within the limits of the system parameters. ISO/IEC 14756:1999-Measurement

and rating of performance of computer-based software systems, provides a guide for rating the performance of a computer-based software system (CBSS) by comparing calculated performance values with the user's requirements and some referenced values [2]. However, new paradigms for processing and storing data, such as cloud computing (CC), require new approaches and models for measuring system performance.

According to ISO/IEC 17788:2014-Cloud Computing-Overview vocabulary [3], Cloud Computing (CC) is a paradigm for enabling network access to a scalable and elastic pool of shareable physical or virtual resources with self-service provisioning and administration on-demand. The cloud computing paradigm is composed of key characteristics, *i.e.* cloud computing roles and activities, cloud capabilities types, cloud deployment models and cloud services. In addition, ISO/IEC 17789:2014-Cloud Computing-Reference architecture [4], mentions that cloud services are grouped into categories, where each category is a group of services that process a common set of qualities. The cloud services in these categories can include capabilities from one or more of the cloud capabilities types such as application capabilities, platform capabilities or infrastructure capabilities, and cloud services categories include: Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS) and Network as a Service (NaaS). Moreover, CC features include easy access to the above services from any location through the Internet, high scalability of resources and services, and the provision of resources shared between consumers as well as dedicated resources.

To improve the quality of CC services, major providers must be able to offer services with Quality of Service (QoS). QoS can be assured through a Service Level Agreement (SLA) between the cloud service provider (CSP) and the customer. SLAs vary from one CSP to the next, and in some cases, different customers can negotiate different contract terms with the same CSP for the same service [5]. SLAs are organized around cloud SLA component clauses that define the concepts commonly used in cloud SLAs. One of the most important of these is the Cloud Service Performance Component (CSPC), which is made up of individual components used in a cloud SLA to define the performance level of the cloud service. Examples of individual components include response time, capacity, and elasticity. For each of these components, there are various measurements and measurement methods.

One of the main problems in measuring cloud services performance is the lack of models that can integrate the perspectives of provider, maintainer and customer in the same model. For example, in a typical IaaS such as Amazon EC2, customers can monitor different parameters that show the state of their virtual machine (VM) and provide system performance information such as memory usage, data transmitted, system load, etc. In this same IaaS, the provider will need to know the performance measurement of all VM instances in order to continually meet the SLA requirements. Moreover, business components and

stakeholders are normally different in each type of cloud service: this means that the performance measurement of a service should include specific measures and measurement methods adapted to the different perspectives and then combined in a general performance measurement according to ISO 15939 [1].

This work proposes P2M2C-3D, a three-dimensional Performance Measurement Model for Cloud Computing, which consolidates performance measurement from the perspectives of provider, maintainer and customer for the different types of cloud services. The design of P2M2C-3D is based on the Performance Measurement Framework for Cloud Computing (PMFCC) [6] and QEST nD , an n -dimensional extension and generalization of a software performance measurement model [7].

This paper is structured as follows. Section 2 presents related work, the concepts underlying the measurement of CC service performance, and the n -dimensional extension and generalization of a software performance measurement model. Section 3 presents the design of the proposed three-dimensional Performance Measurement Model for Cloud Computing (P2M2C-3D) and an overview of a measurement process for it, which analyzes the performance of a cloud service consolidating the perspectives of provider, maintainer and customer. Finally, Section 4 presents a synthesis of the results of this research and suggests future work.

2. Related Work and Background

2.1. Performance Measurement in Cloud Computing

Researchers have analyzed CC performance from various viewpoints. For example, Iosup *et al.* [8] analyzed the CC performance for Many-Task Computing (MTC) systems. According to Iosup, scientific workloads often require high-performance computing capabilities, that is, high-performance execution of loosely coupled applications comprising many tasks. Iosup set out to test the hypothesis that today's clouds can execute MTC-based scientific workloads with similar performance and at lower cost than today's scientific processing systems. For this, the author focused on IaaS providers using public clouds not restricted to an enterprise. Iosup selected four public cloud providers (Amazon EC2, Go-Grid, ElasticHosts and Mosso) to perform a traditional system benchmarking in order to provide a first-order estimate of system performance. His main finding was that the computation performance of the tested clouds was low compared to traditional systems of high-performance computing. He also found that while current CC services are insufficient for scientific computing at large, they are a good solution for scientists who need resources quickly and for a short period of time.

Jackson *et al.* [9] analyzed high-performance computing on the Amazon Web Services cloud. The purpose of his work was to examine the performance of existing CC infrastructures and create a mechanism to quantitatively evaluate them. His study focused on the performance of Amazon EC2, as representative

of the current mainstream of commercial CC services, and its applicability to Cloud-based environments for scientific computing. Jackson quantitatively examined the performance of a set of benchmarks designed to represent a typical High-Performance Computing (HPC) workload running on the Amazon EC2 platform. Kaiqi and Perros [10] proposed an approach for studying CC service performance in terms of delivering QoS guaranteed services to computing environments. They derived an approximation method for calculating the probability and cumulative distributions of the response time for deriving propositions and corollaries to answer service performance questions such as: what is the level of QoS that can be guaranteed? or how many customers can be supported for a given percentile of the guaranteed response time when the cloud service is under a given demand load?

Finally, Montes *et al.* [11] proposed a general-purpose cloud monitoring architecture to address the needs of modern cloud infrastructures based on a layered architecture divided into two main components: 1) access monitoring and 2) data gathering and management. For this, the authors developed a model combining a monitoring level and a dual perspective where the first one corresponds to the type of service to monitor (Server, IaaS, PaaS or SaaS) and the second one to the type of perspective (either provider or client). Although the proposal is quite valuable, one of the shortcomings of this monitoring approach is the lack of a data analysis model that would establish indicators for the representation of performance concepts.

These works present interesting methods for CC performance analysis, but mostly their approach is from an infrastructure standpoint and does not consider CC performance factors from a software engineering perspective. In addition, there is a need to define an integrated model of performance measurement that would include the perspectives of provider, maintainer and customer and that could be applied to the different types of CC services.

2.2. Performance Measurement Framework for Cloud Computing

The Performance Measurement Framework for Cloud Computing (PMFCC) [6] is based on the performance analysis scheme shown in **Figure 1**.

This scheme establishes a set of criteria (or characteristics) for analyzing system performance. In this scheme, system performance is typically analyzed using three sub-concepts—responsiveness, productivity, and utilization—and proposes a measurement process for each.

There are several possible outcomes for each service request made to a system, and they can be classified into three categories. The system may: 1) perform the service correctly, 2) perform the service incorrectly, or 3) refuse to perform the service altogether. Moreover, the scheme defines three sub-concepts associated with each of these possible outcomes, which affect system performance: 1) speed, 2) reliability, and 3) availability [12].

Figure 1 presents this scheme, which categorizes the possible outcomes of a

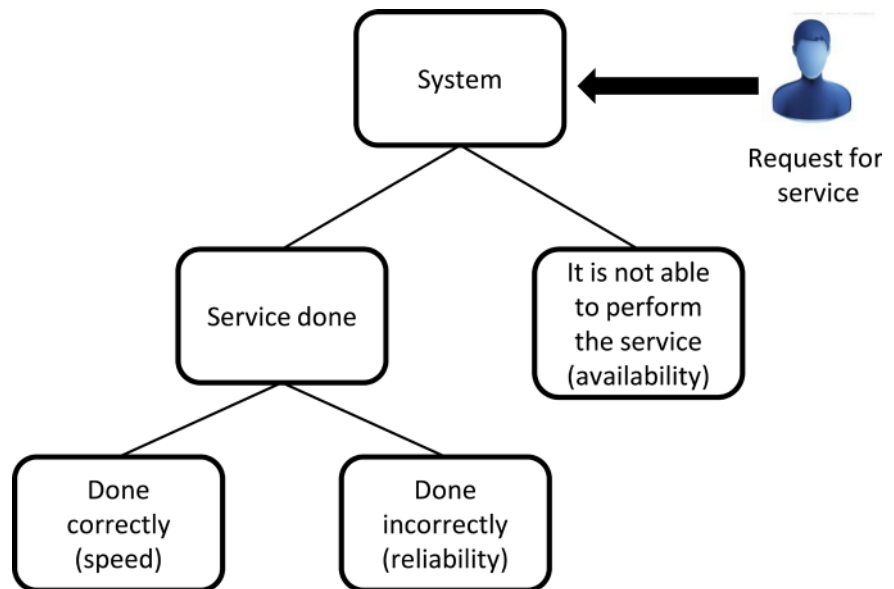


Figure 1. Scheme for performance analysis of a cloud computing system [6].

service request to a system and the sub-concepts associated with them. Based on this scheme, the PMFCC maps the possible outcomes of a service request onto quality characteristics described in ISO/IEC 25010 [13]. This standard defines software product and computer system quality from two distinct perspectives: 1) a quality-in-use model, and 2) a product quality model.

The product quality model is applicable to both systems and software. According to ISO 25010, the properties of both determine the quality of the product in a particular context, based on user requirements. For example, performance efficiency and reliability can be specific concerns of users who specialize in content delivery, management, or maintenance.

The performance efficiency characteristic in ISO 25010 has three sub-characteristics: 1) time behavior, 2) resource utilization, and 3) capacity.

Reliability has four sub-characteristics: 1) maturity, 2) availability, 3) fault tolerance, and 4) recoverability. The PMFCC uses performance efficiency and reliability as measures for determining CC performance.

In addition, the PMFCC proposes the following definition of CCS performance analysis:

“The performance of a Cloud Computing system is determined by analysis of the characteristics involved in performing an efficient and reliable service that meets requirements under stated conditions and within the maximum limits of the system parameters.”

Once the performance analysis concepts and sub-concepts have been mapped onto the ISO 25010 quality characteristics, the framework produces a relationship model (Figure 2) that presents a logical sequence in which the concepts and sub-concepts appear when a performance issue arises in a CC system.

In Figure 2, system performance is based on two sub-concepts: 1) performance efficiency and 2) reliability. As shown in Figure 1, when a CCS receives a

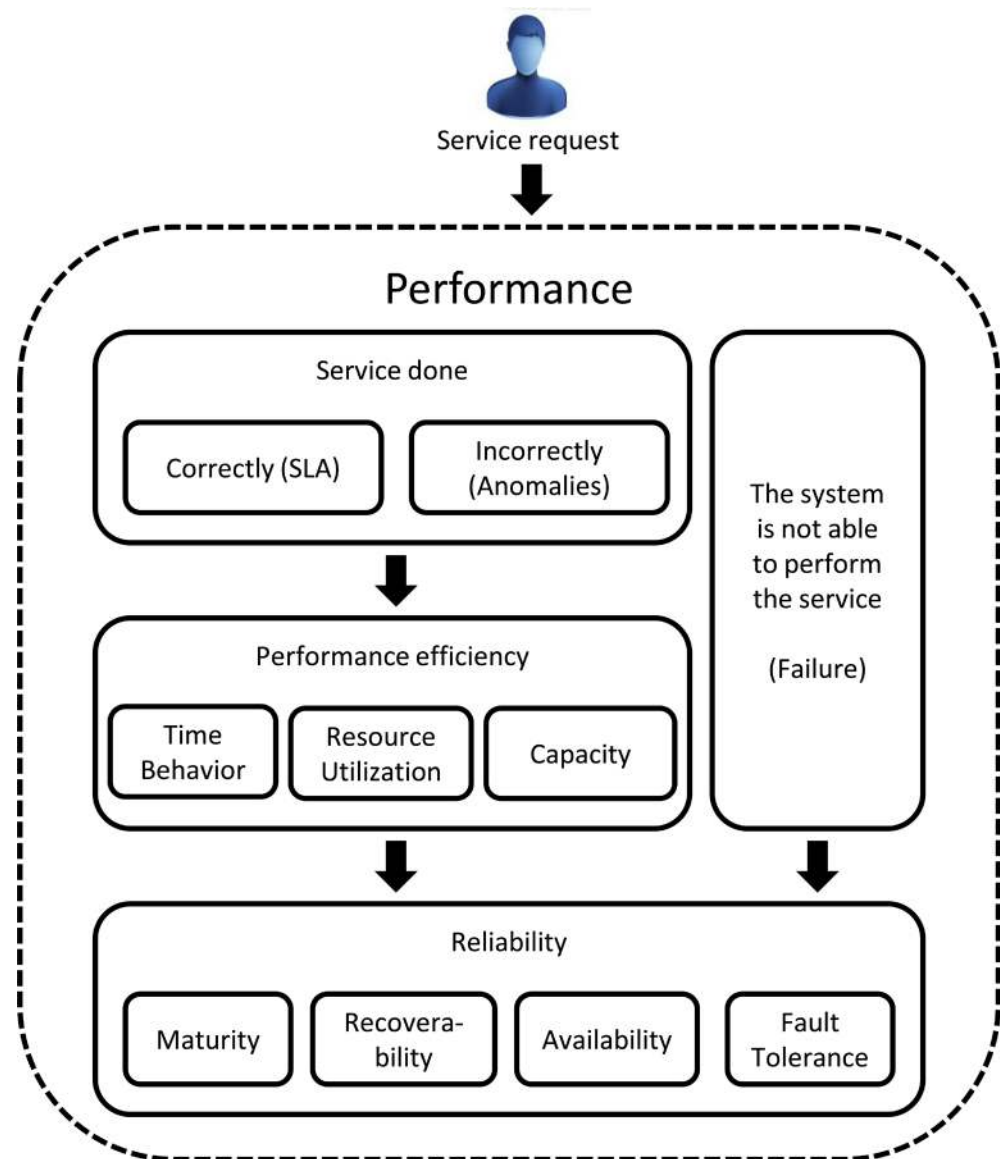


Figure 2. Model of relationship between concepts and sub-concepts involved in performance analysis of a CC service [6].

service request, there are three possible outcomes: the service is performed correctly, performed incorrectly, or cannot be performed. The outcome will determine the sub-concepts that will be applied for performance analysis.

For example, let's assume that during execution the service failed and the request had to be resubmitted; although the service was ultimately performed successfully, it is clear that system availability (a sub-concept of reliability) was compromised, and this affects CCS performance.

2.3. QEST *n*D: *n*-Dimensional Extension and Generalization of a Software Performance Measurement Model

The QEST *n*D model is an extension of QEST (Quality Factor + Economic, Social and Technical dimensions) which takes into account the three stakeholder

categories normally present in the software production process [14]. In some QEST trials, several organizations expressed an interest in taking a greater number of viewpoints into account, which pointed to the need to extend the original three-viewpoint model to one of n possible viewpoints, each to be represented in a nD space resulting from the QEST nD model proposed by Buglione and Abran [7].

QEST nD is a software engineering performance management model that has the ability to handle an independent set of dimensions without predefined ratios or weights. It provides a multi-dimensional structured shell, which can then be filled according to management objectives in relation to a specific project; it can therefore be described as an open model.

The basic purpose of the structured shell is to express performance as the combination of specific measurements (or sets of measurements) selected for each of the n dimensions, these values being derived from both an instrument-based measurement of productivity and a perception-based measurement of quality. **Figure 3** shows the geometrical representation of a tetrahedron as the basis of a QEST 3D model.

Buglione and Abran [14] noted that the tetrahedron occupies a three-dimensional space with a regular triangular base and equal sides like a pyramid. They further noted that the three dimensions (E , S , T) in the space correspond to the corners of the pyramid base, and the convergence of the edges at vertex P represents the top performance level. Therefore, the three dimensions have the same importance in relation to overall performance, which implies that the values for

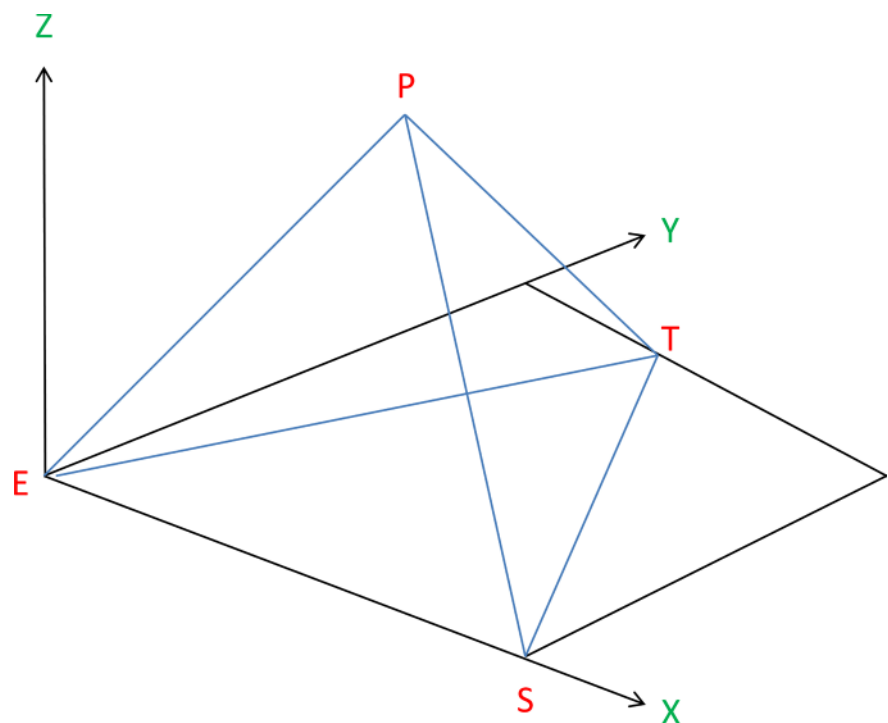


Figure 3. Regular tetrahedron with E , S , T dimensions as pyramid base corners and performance P as vertex [14].

every viewpoint are equal. These values are obtained by normalizing the values of the measurements selected to represent them.

With this 3D geometrical representation, performance can be determined and represented by applying the usual geometrical concepts of distance, area and volume. For example, given a tetrahedron representing a project’s normalized performance level, performance at any point in the project life cycle can be expressed as the ratio between the volume of the lower part of the tetrahedron and its total volume (*minimum* = 0, base of the tetrahedron and *maximum* = 1, apex). **Figure 4** presents the hyperplane sections corresponding to two assessments at different points in time, relative to normalized targets.

3. P2M2C-3D: A Three-Dimensional Performance Measurement Model for Cloud Computing

The PMFCC in Bautista *et al.* [6] defines the basis for a method of measuring Cloud Computing concepts directly related to performance. In this paper, these concepts have been reviewed and updated according to different perspectives as shown in **Table 1**, where the sub-concept measurements are defined according to such perspectives. For example, *TBMP1* is the measurement of time behavior from perspective number one, *i.e.* the provider perspective, while *TBMP2* corresponds to the same measurement from perspective number two (the maintainer perspective) and *TBMP3* corresponds to the customer perspective.

The sub-concept measurements are grouped according to perspective in order to obtain the concept indicator (see **Figure 5**). For example, *MMP1*, *RMP1*,

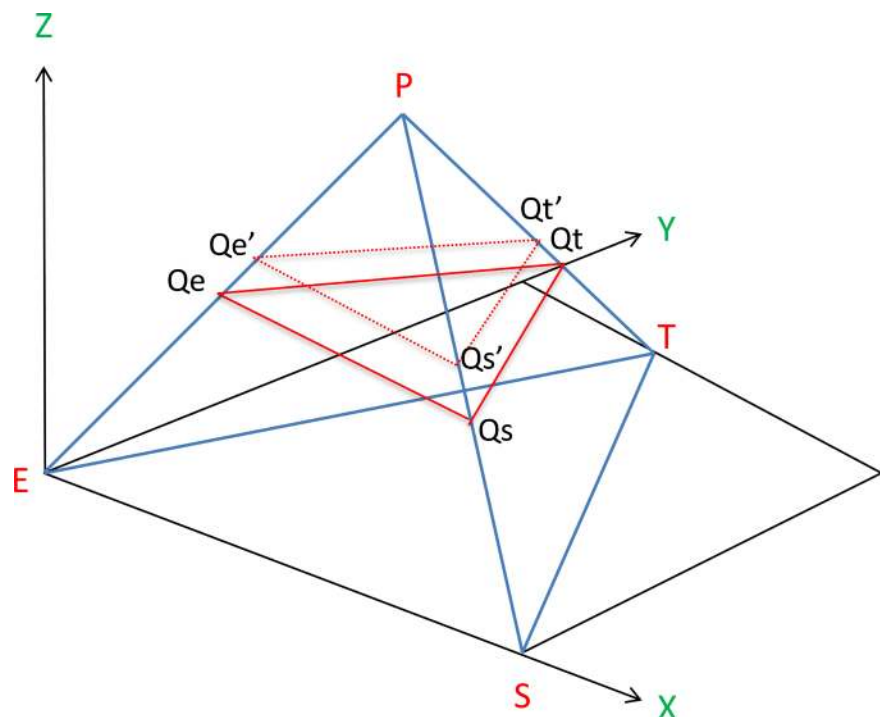
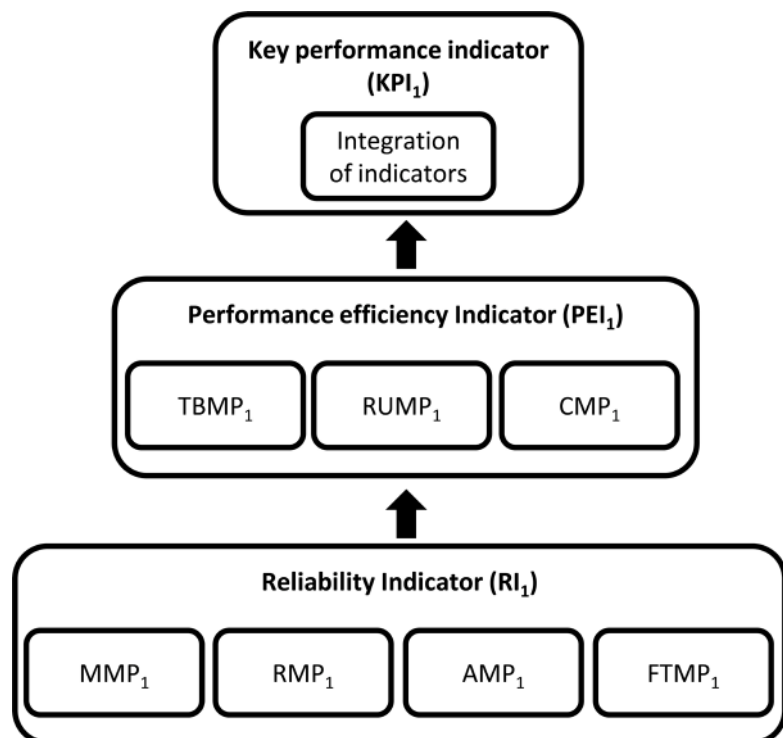


Figure 4. Hyperplane sections corresponding to two assessments at different points in time, relative to normalized targets [14].

Table 1. PMFCC measurements and perspectives.

PMFCC Concept	Sub-concept Measurement	Description
Performance efficiency	TBMP1	Time behavior measurement - provider perspective
	TBMP2	Time behavior measurement - maintainer perspective
	TBMP3	Time behavior measurement - customer perspective
	RUMP1	Resource utilization measurement - provider perspective
	RUMP2	Resource utilization measurement - maintainer perspective
	RUMP3	Resource utilization measurement - customer perspective
	CMP1	Capacity measurement - provider perspective
	CMP2	Capacity measurement - maintainer perspective
	CMP3	Capacity measurement - customer perspective
Reliability	MMP1	Maturity measurement - provider perspective
	MMP2	Maturity measurement - maintainer perspective
	MMP3	Maturity measurement - customer perspective
	AMP1	Availability measurement - provider perspective
	AMP2	Availability measurement - maintainer perspective
	AMP3	Availability measurement - customer perspective
	FTMP1	Fault tolerance measurement - provider perspective
	FTMP2	Fault tolerance measurement - maintainer perspective
	FTMP3	Fault tolerance measurement - customer perspective
	RMP1	Recoverability measurement - provider perspective
	RMP2	Recoverability measurement - maintainer perspective
	RMP3	Recoverability measurement - customer perspective

**Figure 5.** Key Performance Indicators from the provider perspective.

$AMP1$ and $FTMP1$ are combined to obtain the reliability indicator from perspective number one ($RI1$). Similarly, $TBMP1$, $RUMP1$ and $CMP1$ are combined to obtain the performance efficiency indicator ($PEI1$).

The key performance indicator values ($kpi1$, $kpi2$ and $kpi3$) shown in **Figure 6** represent the values of the three dimensions, each placed on its tetrahedron side, describing a sloped plane section (or hyperplane) in the space returning a quantitative assessment which represents the performance at a specific time. Indicators $KPI1$, $KPI2$ and $KPI3$, on the other hand, represent the dimensions of the regular tetrahedron according to the perspectives of provider, maintainer and customer.

3.1. Overview of Measurement Process for P2M2C-3D

As mentioned, each cloud service belongs to a category of services defined in the paradigm of cloud computing. IaaS, PaaS and SaaS provide customers with different types of resources, and for this reason their performance measurements are different and depend on the perspective. Typical offerings are as follows:

- IaaS provides access to computer infrastructure, usually in a virtualized environment.
- PaaS delivers a computing platform and/or solution stack as a service, often through cloud infrastructure in the form of VM instances.
- SaaS delivers software as a service over the Internet, eliminating the need for the end user to install and run the application.

The (IaaS) level is composed of the physical components—servers, networks and storage devices—that the cloud uses to operate. The other levels (PaaS and

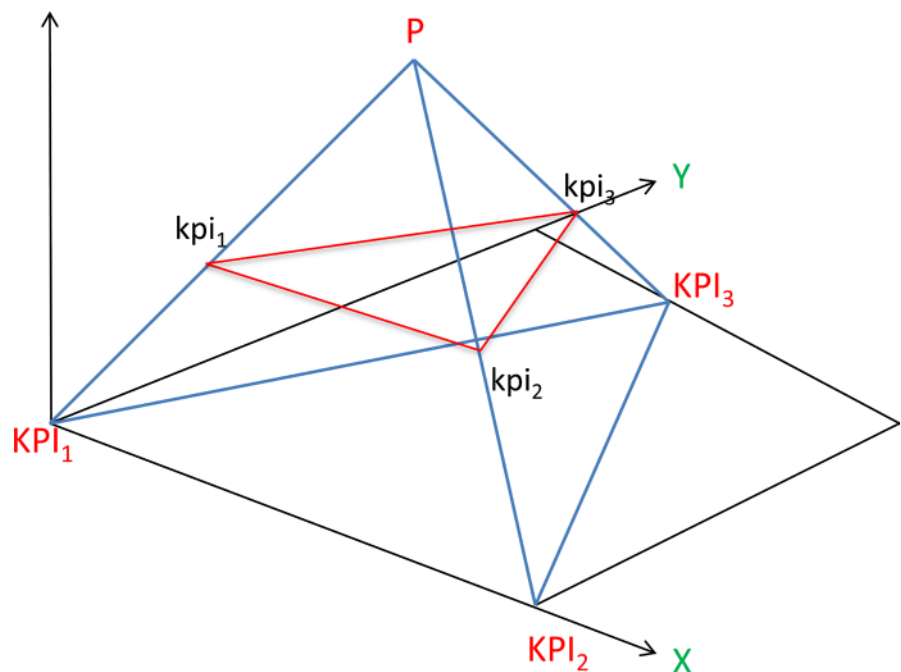


Figure 6. Regular tetrahedron with dimensions $KPI1$, $KPI2$ and $KPI3$, which represent the perspectives of provider, maintainer and customer.

SaaS) contain the virtualized resources and applications to be provided. When considering performance measurement, this differentiation is crucial, since the measuring process for each type of service requires different base measurements. Additionally, each service type has different characteristics, so the measuring methods have to be adapted.

Finally, measurement should provide information about aspects of service performance, and the analysis and use of such information depends not only on what type of service is being measured but also who is obtaining this information and for what purpose. For instance, in a typical IaaS service, maintainers will measure the performance of their virtual machine instances in order to know about system load, memory usage, etc. In the same IaaS, the CSP would need to monitor all VM instances, continuously making sure SLA requirements are satisfied. The CSP would also require measurement data from the infrastructure level (e.g. networking or storage), in order to effectively control the overall system load, VM allocation, migration plans, etc. The kind of information needed therefore depends on which actor (provider, maintainer or customer) is requesting it, and on that actor's role in the cloud ecosystem.

3.2. Selection of Measurements for Different Perspectives

One of the key aspects for implementing the proposed model is the selection of measurements that best represent cloud service performance from the different perspectives. As an example, for an IaaS service, three types of measurements can be distinguished:

- **Provider perspective:** From this point of view, the cloud is regarded as a complex distributed infrastructure, with many hardware and software elements combined to provide a specific set of services. Measurements of this type give the CSP information about the internal performance of those elements. This information serves as an internal status control in order to ensure compliance with SLAs and other service requirements. It can be also used to optimize internal system management and use of resources. Examples of measurements for the provider perspective are *CPU utilization, network TX bytes and HD write errors*, which can be mapped to the concepts of resource utilization, capacity and fault tolerance, respectively.
- **Maintainer perspective:** From this point of view, the cloud is regarded as a system to be managed and maintained, and system views are oriented to its behavior and correct configuration for the customer's objectives in order to provide efficient service. The system maintainer analyzes the performance measurements of every VM instance provided, as well as its internal resources. Examples of measurements for this perspective are *amount of CPU or memory consumed by the VM instances, time of use and storage cost per instance*, which can be mapped to the concepts of resource utilization, availability and capacity, respectively.
- **Customer perspective:** From this point of view, the cloud is regarded as an

abstract entity, capable of providing a specific set of computational services. Performance information provides an abstract description of the cloud service, expressed in the same terms as the service provisioning relationship between the customer and the service (SLA, contract, etc.). This information helps the customer to understand the characteristics of the services received and optimize their use. Measurements that provide useful information and help establish good customer/provider relations include *CC system up-time*, *mean recovery time and response time*, which can be mapped to availability, recoverability and capacity, respectively—concepts that are essential for drafting SLAs and contracts.

Once performance has been measured from the different perspectives, the performance indicators can be determined and represented as shown in **Figure 7**.

Figure 7 shows the key performance indicators for provider, maintainer and customer (kpi_P , kpi_M and kpi_C). The main objective is to have a balanced hyperplane which is as close as possible to the vertex P , which would represent an optimum level of performance from all three perspectives.

4. Conclusion and Future Work

Clouds are highly complex systems designed to provide infrastructure, platform and applications efficiently from any place through the Internet, offering high scalability of resources and services as well as the sharing of resources between consumers. To improve the quality of CC services, major providers must be able to offer QoS. However, one of the main problems in measuring service

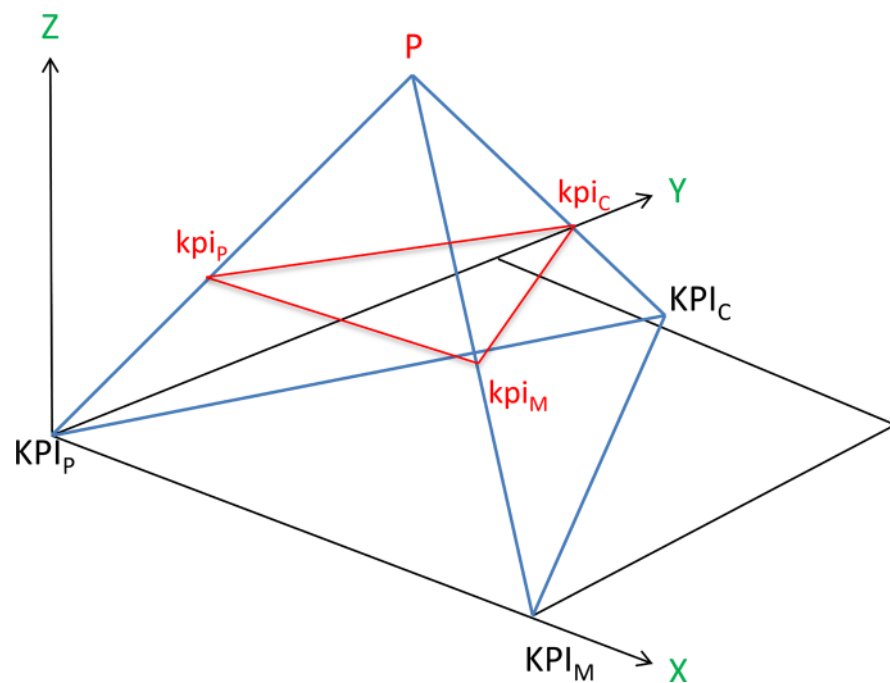


Figure 7. Graphical representation of IaaS performance measurement from provider, maintainer and customer perspectives.

performance is the lack of measurement models that integrate the perspectives of provider, maintainer and customer in the same model. This work proposes P2M2C-3D, a three-dimensional Performance Measurement Model for Cloud Computing, which consolidates measurements from all three perspectives.

Validation of the proposed model will require the following efforts:

1) development of an analysis model and processes for collecting measurements on a specific cloud service from the different perspectives, to be mapped to the performance sub-concepts presented in **Table 1**, and

2) development of a measurement method for obtaining perspective indicators in order to represent the optimal performance level (P) of a cloud service as shown in **Figure 7**.

Future research will therefore focus on the development of measurement collection schemes, methods for obtaining indicators from the different perspectives, and analysis models for the representation of cloud service performance.

References

- [1] ISO/IEC. (2008) ISO/IEC 15939: 2007 Systems and Software Engineering—Measurement Process. International Organization for Standardization, Geneva.
- [2] ISO/IEC. (1999) ISO/IEC 14756: Measurement and Rating of Performance of Computer-Based Software Systems. International Organization for Standardization, Geneva.
- [3] ISO/IEC. (2014) ISO/IEC 17788: Information Technology—Cloud Computing—Overview and vocabulary. International Organization for Standardization, Geneva, p. 10.
- [4] ISO/IEC. (2014) ISO/IEC 17789: Information Technology—Cloud Computing—Reference Architecture. International Organization for Standardization, Geneva, p. 53.
- [5] ISO/IEC. (2015) ISO/IEC Editor's Draft Revised CD 19086-1: Cloud Computing—Service Level Agreement (SLA) framework—Part 1: Overview and Concepts. American National Standards Institute, New York, p. 43.
- [6] Bautista, L., Abran, A. and April, A. (2012) Design of a Performance Measurement Framework for Cloud Computing. *Journal of Software Engineering and Applications*, **5**, 69-75.
- [7] Buglione, L. and Abran, A. (2002) QEST nD: n-Dimensional Extension and Generalization of a Software Performance Measurement Model. *Advances in Engineering Software*, **33**, 1-7. [https://doi.org/10.1016/S0965-9978\(01\)00050-3](https://doi.org/10.1016/S0965-9978(01)00050-3)
- [8] Iosup, A., Ostermann, S., Yigitbasi, N., Prodan, R., Fahringer, T. and Epema, D. (2011) Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing. *IEEE Transactions on Parallel and Distributed Systems*, **22**, 931-945. <https://doi.org/10.1109/TPDS.2011.66>
- [9] Jackson, K.R., Ramakrishnan, L., Muriki, K., Canon, S., Cholia, S., Shalf, J., *et al.* (2010) Performance Analysis of High Performance Computing Applications on the Amazon Web Services Cloud. *IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom)*, Indianapolis, IN, 30 November-3 December 2010, 159-168. <https://ieeexplore.ieee.org/document/5708447/>
- [10] Kaiqi, X. and Perros, H. (2009) Service Performance and Analysis in Cloud Compu-

ting. *World Conference on Services*, Los Angeles, CA, 6-10 July 2009, 693-700.

- [11] Jesús Montes, A.S., Memishi, B., Pérez, M.S. and Antoniu, G. (2013) GMonE: A Complete Approach to Cloud Monitoring. *Future Generation Computer Systems*, **29**, 14.
- [12] Jain, R. (1991) *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. John Wiley & Sons Interscience, New York, NY.
- [13] ISO/IEC. (2011) *ISO/IEC 25010: Systems and Software Engineering—Systems and Software Product Quality Requirements and Evaluation (SQuaRE)—System and Software Quality Models*. International Organization for Standardization, Geneva, p. 43.
- [14] Buglione, L. and Abran, A. (1999) Multidimensional Software Performance Measurement Models: A Tetrahedron-Based Design. In: *Software Measurement*, Springer, 93-107. https://doi.org/10.1007/978-3-663-08949-0_6