Performance measurement for cloud computing applications using ISO 25010 standard characteristics

Anderson Ravanello, Jean-Marc Desharnais, Luis Eduardo Bautista Villalpando,
Alain April, Abdelouahed Gherbi

Department of Software Engineering and Information Technologies
École de Technologie Supérieure, University of Quebec, Montreal, QC, Canada
ravanello@gmail.com, jean-marc.desharnais@etsmtl.net, luis.bautistav@gmail.com,
alain.apri@etsmtl.ca, Abdelouahed.Gherbi@etsmtl.ca

***Abstract:* Measuring the performance of cloud computing-based applications using ISO quality characteristics is a complex activity for various reasons, among them the complexity of the typical cloud computing infrastructure on which an application operates. To address this issue, the authors use Bautista's proposed performance measurement framework** [1] **on log data from an actual data center to map and statistically analyze one of the ISO quality characteristics: time behavior. This empirical case study was conducted on an industry private cloud. The results of the study demonstrate that it is possible to use the proposed performance measurement framework in a cloud computing context. They also show that the framework holds great promise for expanding the experimentation to other ISO quality characteristics, larger volumes of data, and other statistical techniques that could be used to analyze performance.**

Keywords: cloud computing, ISO 25010, SQuaRE, big data, measures.

1 INTRODUCTION:

Cloud computing is complex environment which is dependent on a number of IT infrastructures, including components that are often widely geographically dispersed, with shared elements and running diverse applications [2], [3]. This technology uses hardware and software to deliver ubiquitous, resilient, scalable, billed-by-use, agnostic application systems [4]. There are many advantages [5], [6] and disadvantages [7], [8]**,** [9] to using such a technology, and one of its major disadvantages is addressed in this research: **unreliable system performance due to the complexity of the infrastructure**

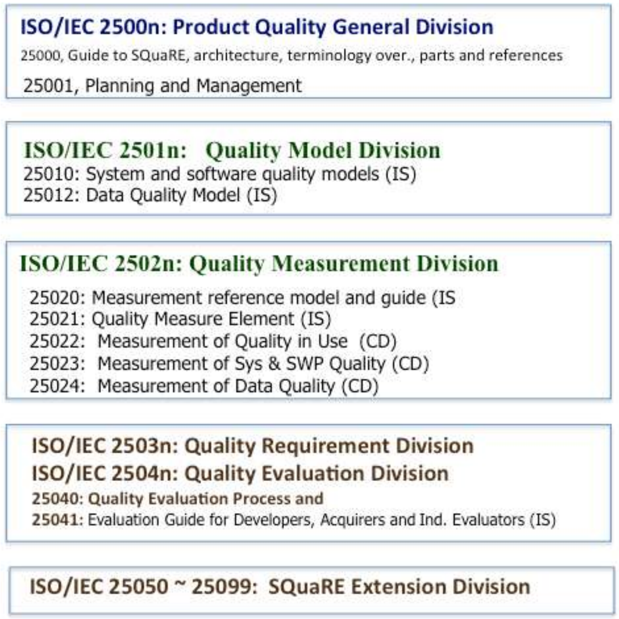
Bautista et al. [1] propose a performance measurement framework for cloud computing systems, which integrates the software quality concepts proposed by ISO 25010, which states: *"The performance of a Cloud Computing system is determined by analysis of the characteristics involved in performing an efficient and reliable service that meets requirements under stated conditions and within the maximum limits of the system parameters."* [12] In their attempt to put the ISO 25010 concepts into practice, the authors found it a challenge to identify and **map the base measures and derived measures (ISO/IEC 25021 and ISO/IEC 15939)** that need to be linked to ISO characteristics and sub characteristics representing the system's behavior.

The objective in this paper is to explain how base and derived measures can be mapped to reveal the performance of a cloud-based application. We have chosen as our case study a large Microsoft Exchange application installed on a private cloud and its distributed clients that run Microsoft Outlook in order to test the proposed Bautista et al. framework on the measurement of unreliable system performance taking into account the complexity of the infrastructure. The results show that we have found a means of mapping base and derived measures, as suggested by ISO/IEC 25010, that simplifies the representation of the cloud computing infrastructure, and also validates the usability of the concepts expressed in the ISO/IEC 25010 standard. This initial research activity will help identify the relevant measures for statistical analysis of ISO quality characteristics in cloud computing applications. This paper focuses only on the concept of time behavior defined in ISO/IEC 25010.

2 ISO/IEC 25010 SOFTWARE QUALITY ASSESSMENT STANDARD

2.1 THE ISO 25000 FAMILY OF STANDARDS (SQUARE)

The Software product Quality Requirements and Evaluation,(SQuaRE) series of standards is composed of many documents destined for many different audiences. It is made up of 5 groupings and 14 documents, the most important of which are shown in Figure 1: Quality Management (ISO/IEC 2500n), Quality Model (ISO/IEC 2501n), Quality Measurement (ISO/IEC 2502n), Quality Requirements (ISO/IEC 2503n), Quality Evaluation (ISO/IEC 2504n) and its Extensions (ISO/IEC 25050 - 25099).



ISO/IEC 2500n: Product Quality General Division
25000, Guide to SQuaRE, architecture, terminology over., parts and references
25001, Planning and Management

ISO/IEC 2501n: Quality Model Division
25010: System and software quality models (IS)
25012: Data Quality Model (IS)

ISO/IEC 2502n: Quality Measurement Division
25020: Measurement reference model and guide (IS
25021: Quality Measure Element (IS)
25022: Measurement of Quality in Use (CD)
25023: Measurement of Sys & SWP Quality (CD)
25024: Measurement of Data Quality (CD)

ISO/IEC 2503n: Quality Requirement Division
ISO/IEC 2504n: Quality Evaluation Division
25040: Quality Evaluation Process and
25041: Evaluation Guide for Developers, Acquirers and Ind. Evaluators (IS)

ISO/IEC 25050 ~ 25099: SQuaRE Extension Division

Figure 1: Five document groupings[1].

2.2 ISO 25010 CHARACTERISTICS AND SUB CHARACTERISTICS

ISO 25010 describes three different quality models for software products: 1) quality-in-use model; 2) product quality model; and 3) data quality model. Each of these models proposes different quality characteristics to represent the quality concepts required to assess software performance from the various perspectives. The first of these, the quality-in-use model, which is designed to measure the quality of software from a user's perspective, proposes five characteristics: effectiveness, efficiency, satisfaction, freedom from risk, and context coverage. The second, the software product quality model, proposes eight characteristics: functional suitability, performance efficiency, compatibility, usability, reliability, security, maintainability, and portability. (We do not consider the data quality model in this paper.)

The main challenge in assessing quality in use and the quality of a software product is to answer the following questions:

- What are the best characteristics and sub characteristics for evaluating the quality of the system to be measured?
- Which derived measures will help in evaluating the quality of the system to be measured based on the characteristics and sub characteristics selected?
- Which measures can be used to form the basis of the derived measures?

In the next section, we explain the concepts of the base measure and the derived measure, as defined by the ISO.

2.3 Base and derived measure concepts (ISO 15939 and ISO 25021)

A base measure is "a measure defined in terms of an attribute and the method for quantifying it" [10], and is derived from a **measurement method**. The same definition is proposed in the International Vocabulary of Basic and General Terms in Metrology [11]. A derived measure is a measure that is defined as a function of two or more values of base measures [12, 13]. This definition is an adaptation of the one in the International Vocabulary of Basic and General Terms in Metrology[2] [11] [14] [15]. A measurement method is defined as a logical sequence of operations, described generically, which is used in quantifying an attribute with respect to a specified scale [14]. It is also based on the definition in [11].

2.4 CLOUD COMPUTING REFERENCES AND STANDARDS

According to the cloud computing (CC) study group of ISO subcommittee 38, CC "is a paradigm for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable cloud resources accessed through services which can be rapidly provisioned and released with minimal management effort or service provider interaction" [15]. This technology employs hardware and software to deliver ubiquitous, resilient, scalable, billed-by-use, agnostic application systems (Prasad, 2010). CC service technology is often subdivided into four different service categories, where each category possesses a common set of qualities. The Cloud service categories are: 1) Infrastructure as a Service (IaaS); 2) Platform as a Service (Paas); 3) Software as a Service (SaaS); and 4) Network as a Service (NaaS). Services in each category may include capabilities from one or more cloud

[1] (IS) means that the document is already a standard, and (CD) means that the document is in the process of being written or rewritten.

[2] The International Vocabulary of Basic and General Terms in Metrology (VIM) is the standard used to define unit measures in science (e.g. meter, degree Celsius, etc.).

capability types [16]. These service models can be hosted and managed in-house or offered by third-party providers. In this first research case study, the CC infrastructure analyzed is a Private Cloud [17] [18] [19].

In addition to the definition of CC provided by ISO subcommittee 38, another frequently cited definition of CC is the one provided by the US National Institute of Standards and Technology (NIST): "*Cloud computing is a model for enabling convenient, on demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction*" [20].

From these definitions, we can see that CC is a technology that relies on the connectivity provided by the Internet to allow easy access to shared pools of resources, and withdrawal from those shared pools without much administrative effort. In principle, these shared resources allow a high degree of flexibility to process variable workloads and can be managed via a service level agreement (SLA) describing the expected behavior and performance of a CC service. From the SQuaRE perspective, the quality in use of a CC service is directly linked to the quality of the network infrastructure, as well as to the configuration of its pooled resources.

While there are a number of advantages to using the CC technology, such as the flexibility to add new resources to respond to consumption peaks, it is important to point out that CC is not the solution to all computing problems. There are a few caveats, one of which is that the infrastructure can become so complicated, with many nodes and inter-node relationships, that its time behavior is unreliable. This unreliability is one of the concerns that we address in this paper.

3 Case study

Our case study was conducted on commercial software running on the CC infrastructure of a private cloud **that mainly hosts and enables access to a company's email services.** Organizations devote considerable administrative effort to retaining control of as much of their private cloud infrastructure as possible. A simplified version of the more complex infrastructure of the CC network in this study is represented in the high level network map depicted in Figure 2. The 12 nodes (or steps) of the network are shown, along with the information flow between the users and the email service..

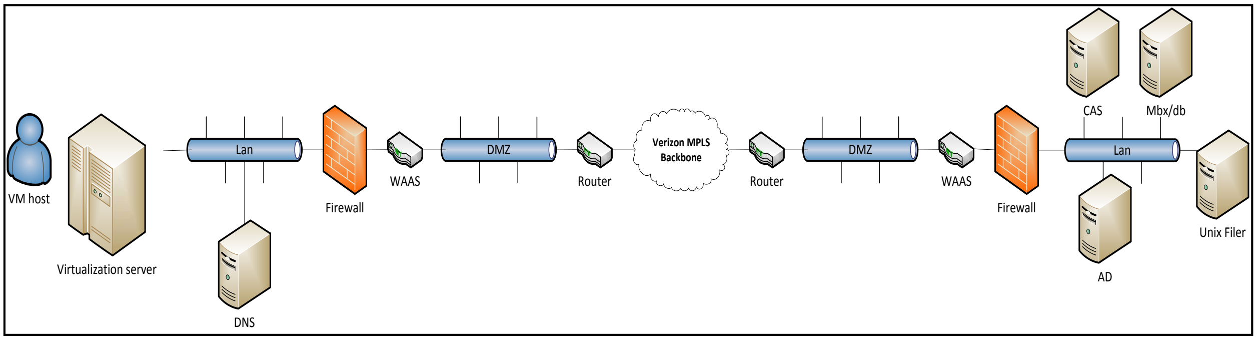


Figure 2: High level network map of a Cloud Computing application for an email service (from the authors).

These 12 nodes are the following:

1) VM Host (Virtual Machine Host): the server, either physical or virtual, that provides computing resources to one or more virtual machines

2) Virtualization server: a virtual host capable of creating a local cluster of virtual machines

3) DNS (Domain Name System): the local server for name resolution

4-8) Firewall, DMZ (demilitarized zone): software components associated with gateways that provide security and communication between a computer system and the Local Area Network (LAN) or Wide Area Network (WAN)

9) CAS (Client Access Server): the server cluster that receives, queues, and processes user requests for data

10) MBX/db (Mailbox database): a database server that contains, organizes, and indexes user data

11) AD (Active Directory): the services that handle security, authorization, and name resolution on the data center side of the system

12) Unix Filer: a file server for the storage, backup, and archiving of high volume user data not handled by the MBX/DB.

Simplification of the network map representation involved removing items like the individual physical servers and virtual machines that make up the server clusters. This was done to highlight a broader concept of tasks, such as those performed by nodes labeled CAS or AD. In addition, only one network path is represented in Figure 2, which is not always the case in real situations. For example, while a single transaction will always flow via a single channel, multiple channels will often be available for transmitting transactions, such as fault tolerant communication links, switches, firewalls, and routers. Finally, the host machine is represented here unconditionally as either a physical or a virtual machine.

As indicated in section 2.4, many measures are captured by the software that makes up this CC infrastructure. Most of these measures are considered derived measures from an ISO point of view, because they use at least two base measures. The derived measures that we use in this case study are the most atomic and granular measures we have found and that are available in operational systems. Consequently, we have labeled them "low level derived measures" in the tables and appendices that follow.

For the network map presented in Figure 2, 159 different base and low level derived measures have been identified as potential candidates to be used to reflect the end-user experience of the performance of the e-mail service, affecting one or more of the nodes. The authors have elected to present only a small sample of the 159 different base and low level derived measures in Table 1, owing to the space and format restrictions of the paper.

Table 1: Subset of base and low level derived measures used in this case study (from the authors)

Host	Virtualization	Lan
Logicaldisk\% free space	Logicaldisk\% free space	IP\Frag Fails
Logicaldisk\% Idle time	Logicaldisk\% Idle time	Network\ Tx Queue Len
Processor \ %processor time	Processor \ %processor time	Network \ Total bytes /sec
System \ threads	System \ threads	Network \ Tx dropped
Logicaldisk\Avg. Disk Bytes /Read	Logicaldisk\Avg. Disk Bytes /Read	Network \ Tx Error
Logicaldisk\Avg. Disk Bytes /Write	Logicaldisk\Avg. Disk Bytes /Write	Network\% cpu utilization
Processor \ Interrupts/sec	Processor \ Interrupts/sec	IP\Hdr Error
Processor\Total Handle count	Processor\Active threads	Current concurrent connections

Table 2 presents the base and low level derived measures analyzed in this case study with the sole objective of measuring the time behavior characteristic (as defined by ISO 25010) of the email service on this private cloud. Table 2 has a column labeled Evaluation, which shows what is considered as the expected behavior according to industry standards. If the column shows 'low', then the lowest value is better. If it shows 'high', then the highest value is better. As explained in section 4, for this paper, the authors have selected only the measures that are evaluated as LOW. Future research will address both HIGH and LOW measures, as in section 6.

It should be noted that, even though these measures are derived, it is possible to capture them in the email service using commercially available tools. It is also important to mention that these measures[3] need to be related to the characteristics present in the ISO 25010 model, and that they cannot be bundled or compared, because, in some cases, they do not relate to the same characteristic nor do they compared easily one—by-one. We decided to arrange each base measure on one of two axes, in order to facilitate association of the base measures with the characteristics of ISO standards, that is, **vertical axis** and **horizontal** axis measures.

Vertical axis measures interrelate on a per-node basis to influence time behavior. We can see from Table 2, for

[3] The term "measure" means "low level derived measure" in this paper, unless otherwise stated.

example, that memory utilization, processor time, and disk space can all influence the time behavior of the email service as perceived by the end user, and they are all based on the same node.

The following are the defining characteristics of the vertical axis measures that we have chosen:

a) They are context-independent: Vertical measures will always fluctuate in relation to the workload on the node that is being analyzed;
b) They are interrelated: Vertical measures, when analyzed from the perspective of multivariate statistics, should always present a high degree of colinearity, while meeting evaluation expectations;
c) They are timely indicators: Vertical measures serve as good points of comparison between different time stamps, and, as they are interdependent and unaffected by context, they can be analyzed to identify variations whenever end-user performance fluctuates.

Horizontal axis measures represent the base and low level derived measures associated across the nodes of the network map. They always affect the system as a whole, and have the potential to show the system "as is", and are not such precise historical measures as the vertical measures.

The following are the defining characteristics of the horizontal axis measures we have chosen:

a) They are context-dependent: The analysis of each grouping of measures only makes sense in the context that is being explored;
b) They are self-sufficient: Each horizontal measure can influence one or more performance characteristics;
c) They are atemporal: : Horizontal measures cannot be applied to events that occur during specific time frames, and so they cannot be included in comparisons such as "historical averages".

In this case study, we implement the horizontal axis measures, and propose a usage index that will be further discussed in sections 4 and 5.

Table 2: Base and low level derived measures and sub characteristics of time behavior on the cloud studied (from the authors)

Sub characteristic	Evaluation	Base and low level derived measures
Transmission	Low	CAS.System\Processor Queue Length
	Low	CAS.Web Service\Bytes Sent/sec
	Low	CAS.Network Interface\Network Bites
	Low	HOST.Outlook\RPCs attempted – UI
	Low	HOST.Process (Outlook)\IO Read Bytes/sec
	Low	HOST.Process (Outlook)\IO Write Bytes/sec
	Low	HOST.System\threads
Time	Low	HOST.Logicaldisk\Avg. Disk Bytes/Read
	Low	HOST.Logicaldisk\Avg. Disk Bytes/Write
	Low	HOST.Network Interface\Bytes Sent/sec
	Low	HOST.Outlook\Time AVG (200)
	Low	HOST.Physical Disk\Disk Read Bytes/sec
	Low	HOST.Physical Disk\Disk Write Bytes/sec
	Low	HOST.Process (Outlook)\Page Faults/sec
	Low	HOST.Processor\%Processor Time
Task	Low	CAS.Web Service\Connection Attempts/sec
	Low	CAS.Web Service\Current Connections
	Low	MBX/DB.Network Interface\Bytes Received/sec
	Low	HOST.Outlook\RPC UI Shown
	Low	HOST.Process (Outlook)\% Privileged Time
	Low	HOST.Process (Searchindexer)\% Privileged Time
	Low	HOST.Process (Searchprotocolhost)\% Privileged Time

4 METHODOLOGY

In this section, we describe the five phases we followed in performing the case study: 1) data collection; 2) choice of ISO characteristics and sub characteristics; 3) data organization; 4) data analysis; and 5) data presentation.

4.1 DATA COLLECTION

The data analyzed in this case study were collected automatically as part of the standard operational system data collection process for the private cloud. They are stored in LOG format for two different platforms, one for client machines and the other for servers, for future use. They can also be resorted for troubleshooting purposes.

The private cloud in this case study is composed of approximately 80,000 client machines and 10,000 servers and network devices. With this number of clients and servers, around 800,000 data points per minute are generated. To visualize this situation, imagine a spreadsheet which grows by 800,000 lines of data every second, with each line made up of approximately 1,000 characters.

This represents a rate of data collection of 1.9 GB per minute. For this paper, only 600 MB of these data has been studied – see Figure 3.

4.2 CHOICE OF ISO CHARACTERISTICS AND SUB CHARACTERISTICS

Using ISO/IEC 25010 (see section 2), it is possible to associate the vertical and horizontal measures to quality characteristics and sub characteristics proposed in this standard. With the data available in this case study, we were able to assess the **time behavior** quality characteristic via the email service usage measures presented in Table 2. For convenience, in this paper, we used only those measures evaluated as LOW, mainly because there more of these than there are measures evaluated as HIGH.

4.3 DATA ORGANIZATION

This phase of the case study is concerned with the organization of the dataset. Initially, we chose to divide the data collected according to their three physical locations (North America, Europe, and Asia), and processed over a 1 week period during business and non business hours. A sub grouping of the data was created per host status (machine level) to allow us to compare usage on a machine by machine basis. In addition, in order to determine the time behavior of the whole system, a data grouping was created to represent the overall status of the nodes analyzed in the form of various indices and graphs, as presented in section 5. This data grouping was derived from Bautista's framework [1] and is partially implemented in this paper.

4.4 DATA ANALYSIS

The data were analyzed by observing two different sets of evidence: statistical analysis results, and the results of a visual examination of the time behavior graphs. With the statistical analysis results, we were able to compare the vertical and horizontal measures numerically. For this comparison, we looked at four values: averages, variances, kurtosis, and skewness.

We studied the time behavior graphs in two different formats: time lines and radial graphs. The time lines gave us an understanding of the cloud cycles, and the radial graphs enabled us to easily identify the groupings of the peaks. The time frame we considered for the radial graphs was 8 am to 6 pm, even though the company operates on the more typical 9 am to 5 pm timetable. This time frame (8am to 6pm) helped on identifying the beginning and end of the work cycles in work hours. One of these graphs is depicted in Figure 3 (small values are better).

It is important to note that, because of the number of data to be considered in these analyses, merely performing them has an impact on system performance.

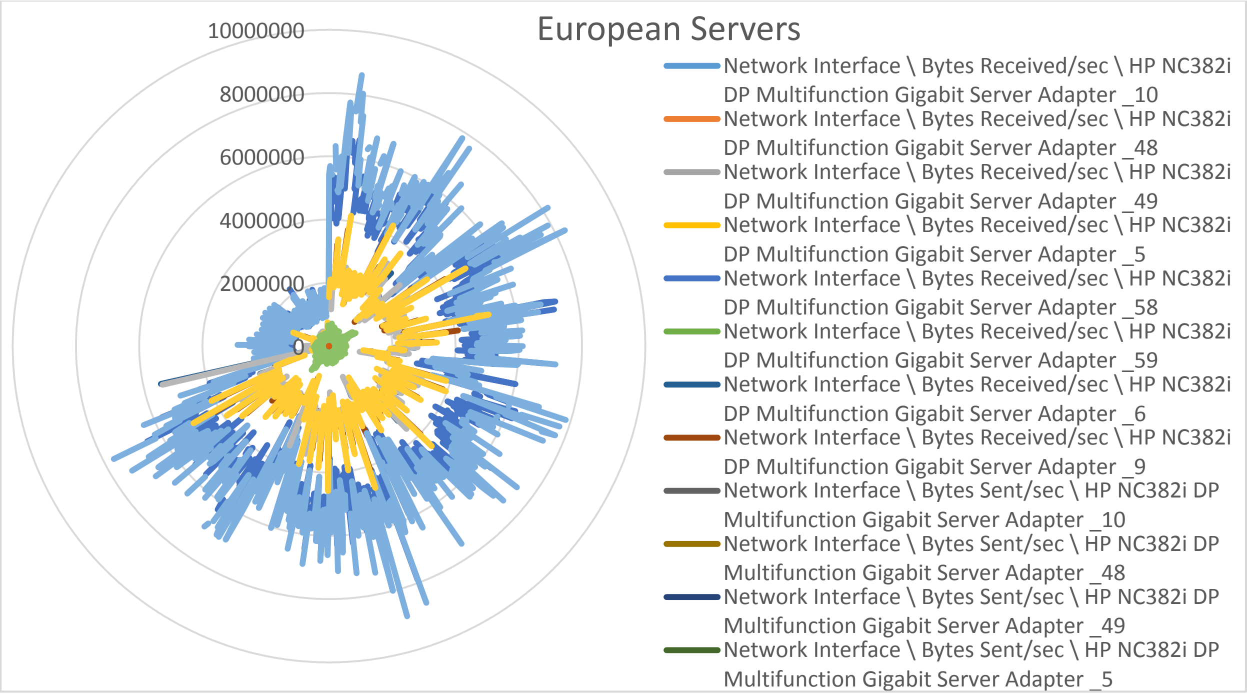


Figure 3: Example of a radial graph – European servers, 8 am to 6 pm.

Figure 3 shows the measures considered, each of which is represented by a different color, as well as the interrelationships of the measures at the same point in time. For example, **light blue** and **dark blue** represent different measures, but they are highly coincidental. The non business hours are located in the upper left quadrant of the image. As fewer data points are generated outside business hours, fewer points are shown in the graph for those hours.

4.5 Data presentation

One of the objectives in measuring the time behavior of the email service is to help with the administrative challenges of making decisions and taking action on the CC infrastructure when needed. With that requirement in mind, the radial graph shows the evolution of the time behavior measures that contribute to the management of the email service.

The significance of the data is determined by measuring the area of the radial graph: the **larger** the graph, **the worse** the time behavior. Data collectors are presented in Figures 3 and 4.

5 Analysis

The objective of the case study is to try to measure the time behavior of the email service operating on a CC infrastructure. We tried to restrict the amount of resources consumed in collecting measures using a performance log created by the commercially available tools that we use daily. Statistical evidence is corroborated by empirical knowledge, to the effect that the more resources are consumed, the worse the time behavior of the email service. This can be easily seen by comparing business hours and non business hours in Figure 3.

Table 3 provides a sample of the statistical analysis, which suggests that business hour skewness is positive. This shows that, in general, the users of the email service tend to exhibit worse than average time behavior during business hours, because the measured characteristics demonstrate better behavior on **lower values**. There is also one high Kurtosis value, which suggests that, wherever the peaks occur, they are high in comparison with the measured average. The levels of variance are, in all cases, quite high relative to the mean, suggesting an abnormal distribution.

Table 3: Base statistics for one measure, business hours (ON) vs. non business hours (OFF) (from the authors)

Web Service \ Bytes Sent/sec \ _Total (On hours)	*Web Service \ Bytes Sent/sec \ _Total (Off hours)*

Mean	43744.48	Mean	28469.07
Standard Error	1589.89	Standard Error	979.83
Median	20608.22	Median	18614.77
Mode	#N/A	Mode	#N/A
Standard Deviation	84981.26	Standard Deviation	35450.64
Sample Variance	7221815159	Sample Variance	1256748343
Kurtosis	27.77	Kurtosis	49.97
Skewness	5.01	Skewness	5.73
Range	821989.44	Range	428426.46
Minimum	2925.47	Minimum	3169.28
Maximum	824914.92	Maximum	431595.75
Sum	124978002.8	Sum	37266022.34
Count	2857	Count	1309
Largest (5)	691174	Largest (5)	343377.62
Smallest (5)	3205.38	Smallest (5)	3214.31
Confidence Level (95.0%)	3117.455376	Confidence Level (95.0%)	1922.226166

This example helps to illustrate that “average usage” isn’t a very good measure to use in analyzing the ISO’s Time Behavior quality characteristic. These initial statistical elements also make it possible to compare single nodes with the whole infrastructure, which helps to determine possible points of failure of the CC infrastructure. These findings point to a need for other statistical methods, to be investigated in future case studies.

Considering the many factors involved in the analysis of a CC environment, we will next consider the design of a collector for the horizontal measures. This will allow a general, multi node observation of the cloud that should make it possible to identify single moments of peak resource usage, as well as a grouping of averages. Figure 3 shows a graph with a complete data series, while Figure 4 shows only a single moment in time. Finally, the evolution of the index over time is illustrated in Figure 5.

The time behavior in this time series shows a peak in usage early on Tuesday and another at the end of the day on Friday. This situation should be further investigated and interpreted using representations similar to those in Figures 3 and 4.

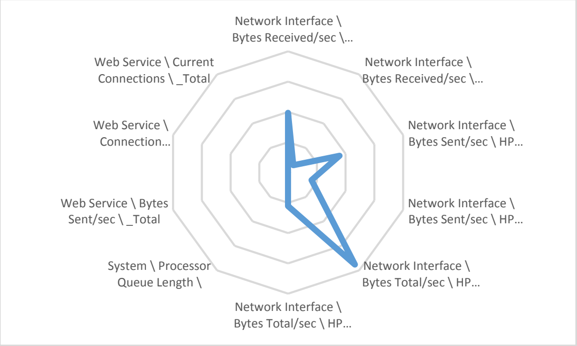


Figure 4: Time behavior at a single moment in time.

On the graph in Figure 4, we can see that a number of low level derived measures are activated in a single moment in time. This should help us to further refine the graph presented in Figure 3.

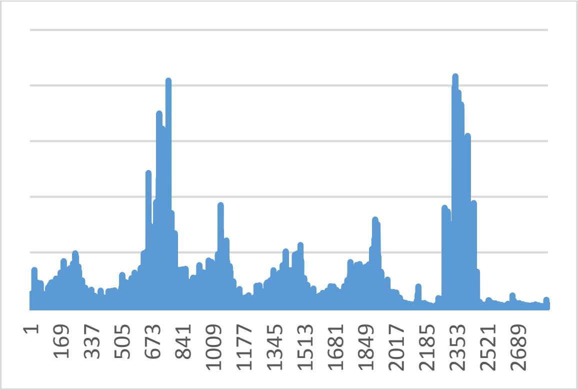


Figure 5: Time behaviour index from Monday at 12 am to Sunday at 11 pm.

The index illustrated in Figure 5 was generated by measuring the areas of each of the moments in time (see Figure 4 for an example of the graphical representation of a single moment). It shows peaks on Tuesday morning and Friday afternoon. These peaks could be further investigated, using the index to nalyze the time behavior graphs for each of the peak occurrences. In this graph, every number on the X axis corresponds to one data point. This representation can be used to help identify which data point should be analyzed for each peak; for example, 720 and 2418 for this case study.

6 Conclusion and Future Research

This paper demonstrates that it is possible to organize measures currently available in the performance log files of a typical data center using a CC infrastructure, and use them to show the time behavior quality characteristic as defined in the ISO's Software product Quality Requirements and Evaluation (SquaRE) series of standards following Bautista's proposed quality measurement framework. Three main challenges become apparent as a result of this case study analysis: collecting, processing, and representing the data.

Data collection from several hosts is a complicated issue because of the amount of data generated and the resources consumed during the generation process. Also, analyzing time behavior may affect the quality of the system, simply because of the additional overhead as a result of collecting, processing, transmitting, and storing the data. The processing for this empirical study was performed ex post facto in an offline database. Processing the same amount of data during production time would be an interesting computational challenge. The representation of multivariate data is a challenge in itself, with radial graphs proposing an interesting solution. However, these graphs are not a widely known form of data representation, which contributes to the difficulty inherent in performing this step.

Multiple avenues for further research follow from this paper: more of the base and low level derived measures that make up the other quality characteristics of the ISO 25010 model should be used; a longer time frame could be studied; and other statistical methods could be used to give a better answer to the research questions.

Finally, the authors also suggest that more research is required to reinforce the validity of this framework and expand our understanding of it.

7 References

[1] L. Bautista, A. Abran and A. April, Design of a Performance Measurement Framework for Cloud Computing, Journal of Software Engineering and Applications, vol. 5, no. 2, 2012, pp. 69-75, 2012.

[2] N. Mirzaei, Cloud Computing, Institute Report, Community Grids Lab, Indiana.Edu, 2008.

[3] S. Sivathanu, Y. Mei, L. Liu and X. Pu., Performance Measurements and Analysis of Network I/O Applications in Virtualized Cloud, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.165.6742&rep=rep1&type=pdf, 2010.

[4] B. R. Prasad, E. Choi and I. Lumb, A taxonomy, survey, and issues of Cloud Computing Ecosystems, London: Springer-Verlag, 2010.

[5] M. Creeger, "CTO roundtable: cloud computing," in *Communications of the ACM*, 2009.

[6] N. Phaphoon, N. Oza, X. Wang and P. Abrahamsson, "Does Cloud Computing deliver the promised benefits for IT industry?" *Proceedings of the WICSA/ECA*, pp. 45-52, 2012.

[7] M. Armbrust and A. Griffith, "A Berkley View of Cloud Computing," UCB/EECS-2009-28 EECS Department, UCB, Berkley, CA, 2009.

[8] N. Gruschka and M. Jensen, "Attack surfaces: a taxonomy for attacks on cloud services," *Proceedings of the 3rd IEEE International Conference on Cloud Computing*, pp. 276-279, July 2010.

[9] B. Grobauer, T. Walloschek and E. Stocker, "Understanding Cloud Computing Vulnerabilities," *IEEE Security and Privacy*, vol. 9, no. 2, pp. 50-57, 2011.

[10] ISO/IEC, ISO/IEC 15939:2007, Systems and Software Engineering, Measurement process, 2007.

[11] JCGM, International Vocabulary of Metrology – Basic and General Concepts and Associated Terms, 2006.

[12] ISO/IEC 25010:2010 SOFTWARE ENGINEERING – SOFTWARE PRODUCT QUALITY REQUIREMENTS AND EVALUATION (SQUARE) – SYSTEM AND SOFTWARE QUALITY MODELS,, ISO/IEC JTC 1/SC 7, 2010.

[13] ISO/IEC 25021:2012 Software Engineering – Software product Quality Requirements and Evaluation (SQuaRE – Quality Measure Elements, ISO/IEC JTC 1/SC 7, 2012.

[14] J. Desharnais and A. Abran, "Software Measurement Methods: An Analysis of Two Designs," *Journal of Software Engineering and Applications,* pp. 797-809, 10 2010.

[15] ISO/IEC JTC 1 SC38:Cloud Computing Overview and Vocabulary, Geneva, Switzerland: International Organization for Standarization, 2012.

[16] ISO/IEC, ISO/IEC, ISO/IEC DIS 17789: Information technology - Cloud Computing - Reference Architecture, Geneva, Switzerland: International Organization for Standardization, 2013.

[17] A. Iosup, S. Ostermann, Y. Nezih, R. Prodan, T. Fahringer and D. Epema, "Performance Analysis of Cloud Computing Services for Many-Task Scientific Computing," *IEE TPDS Many-Task Computing,* November 2010.

[18] Y. Mei, X. Pu and S. Sivathanu, "Performance Measurements and Analysis of Network I/O Applications in Virtualized Cloud," Georgia Institute of Technology, Atlanta, GA, 2010.

[19] S. Suakanto, S. Supangkat, S. Saragih and R. Saragih, "Performance MEasurement of Cloud Computing Services," *International Journal on Cloud Computing: Services and Architecture (IJCCSA),* p., vol. 2, no. 2, April 2012.

[20] NIST – National Institute of Standards and Technology, "The Nist Definition of Cloud Computing," CSD - TIL - NIST, Gaithersburg, MD, 2011.

[21] ISO/IEC, ISO/IEC 25000:2005 Software Engineering -- Software product Quality Requirements and Evaluation (SQuaRE) -- Guide to SQuaRE, ISO/IEC JTC 1/SC 7, 2005.

[22] K. D. Gotzamani, "The implications of the new ISO 9000:2000 standards for certified organizations – A review of anticipated benefits and implementation pitfalls," *International Journal of Productivity and Performance Management,* pp. 645-657, 2005.

[23] W. A. Shewhart, Economic Control of Quality of Manufactured Product, American Society for Quality Control, 1980.

[24] W. E. Deming, Out of the Crisis, Princeton: The MIT PRess, 2000.

[25] A. Feigenbaum, Total Quality Control, McGraw-Hill Companies, 3 Rev Sub edition (January 1, 1991), 1991.

[26] ISO/IEC, TR 9126-(1-4) Software Engineering - Product Quality, Geneva: International Organization for Standardization, 2003.

[27] A. Sousa-Poza, M. Altinkilink and C. Searcy, "Implementing a Functional ISO 9001 Quality Management System in Small and Medium-sized Enterprises," *International Journal of Engineering,* pp. 220-228, 2009.

[28] L. Cagnazzo, P. Taticchi and F. Fuiano, "Benefits, barriers and pitfalls coming from the ISO 9000 Implementation: the impact on business performances," *WSEAS Transactions on Business and Economics,* pp. 311-321, 2010.

[29] B. Poksinska, J. Kahlgaard and M. Antoni, "The State of ISO 9000 Certification: A study of Swedish Organizations," *THe TQM Magazine,* pp. 297-306, 2002.

[30] J. Juran and J. De Feo, Juran's Quality Handbook: The Complete Guide to Performance Excellence, McGraw-Hill, 2010.

[31] J. Voss and J. Zhang, "Cloud computing: new wine or just a new bottle?," *IT Professional,* vol. 11, no. 2, pp. 15-17, 2009.