Associating Performance Measures with Perceived End User Performance:

ISO 25023 compliant Low Level Derived Measures

Anderson Ravanello, Luis Villalpando, Jean-Marc Desharnais, Alain April, Abdelouahed Gherbi
Department of Software Engineering and Information Technologies
École de Technologie Supérieure, University of Quebec,
Montreal, QC, Canada
Email: ravanello@gmail.com, luis.bautistav@gmail.com, jean-marc.desharnais@etsmtl.net,
alain.april@etsmtl.ca, Abdelouahed.Gherbi@etsmtl.ca

Abstract— **This paper applies a measurement procedure to predict the degraded state of a private cloud application using only available data center log low level derived measures (LLDM). Our intent is to improve the discussion of service level agreements of a widely used private cloud computing application (i.e. 80,000 users on 600 servers world-wide). In organizations, cloud application performance measuring is often based on subjective and qualitative measures with very few researches to address the large-scale private cloud perspective. Furthermore, measurement recommendations from ISO proposals (i.e. ISO 250xx series, ISO/IEC 15939 and more recently the ISO/SC38-SLA framework) are poorly adopted by the industry, mainly due to the absence of proof of concept and the high degree of complexity associated with implementing the measurement concepts described in these international standards. To try to demonstrate these concepts, the ISO 25010 performance efficiency characteristics are used with a number of LLDMs to model the state of a large private cloud computing application using indicators such as: normal, abnormal, adequate or degraded. This application still cannot be generalized due to its nature as research in progress.**

Keywords- cloud computing; cloud application; SLA; ISO 25010; end user performance measurement

I. INTRODUCTION

Measuring end user performance has been a concern of software engineering researchers since the early 60's [1]. Many experiments have been created, tested and validated [2] [3] [4] based on a survey with the involved users. Surveys have concerning limitations, such as not being good for following trends in real time, not providing good source for cause and effect, having poor timing response, demonstrating low response and being vulnerable to responder bias. To avoid these frailties, some form of automated, user-independent approach would be helpful.

Software systems performance measurement is currently conducted in many ways. One popular approach is to use the data center logs readily available in different operational systems, applications, computers and IT infrastructure components. Logs are binary files that collect data from different components in a system and store this data in a file or database for posterior analysis. Many commercial, open source, and easily accessible tools, are available for collecting, analyzing and generating performance dashboards that present technical measures[1] of different system components that are used by a software [5], [6], [7].

Measuring performance using measures issued from logs can only measure the internal, and very technical, perspectives of an IT system. This is why the end user[2] performance perspective is often inferred, estimated, approximated and even sometimes guessed, based on experience and using data center log data. The resulting measures may affect or not the actual user's perceived performance according to the observer's perspective and experience [8], [9], [10].

Cloud computing operates in complex environments which are dependent on a number of IT infrastructures, including components that are often widely geographically dispersed, with shared elements and running diverse applications [11]. This technology uses hardware and software to deliver ubiquitous, resilient, scalable, billed-by-use, agnostic application systems [12]. There are many advantages [13], [14] and disadvantages [15] to using such a technology, and one of its major disadvantages contemplated in this research is: the unreliable system performance due to the complexity of the infrastructure used by cloud applications.

Considering these challenges, the authors approach this case study with one particular hypothesis: Is it possible to employ existing data center logs to model degraded cloud computing application performance via the monitoring of two sources of data: 1) low level derived measures and 2) the end user's reports to help desk of such degradations?

In order to conduct this case study, the authors follow three methodological steps:

1. Associate user reports of degraded performance with the LLDM;
2. Map the base and low level measures to the quality characteristics proposed by Bautista [16];
3. Perform a lab experiment to model the performance of a real cloud application.

---

[1] In this paper, the substantive "measure" always refer to the ISO 25000 term "low level derived measure "

[2] The end user is one of the stakeholders of the software, typically the one who uses it to perform a task.

II. END USER PERFORMANCE MEASUREMENT

The problem of measuring information system's performance is not new and has been explored by numerous authors. A number of these use available data center tools for measuring the values of low level and derived measures. Different approaches are implemented in available tools: some install agents on the involved nodes that report the measures back to the performance management database [17]; others monitor the measures via SNMP [3] [18], collecting the measurements directly and other tools store the measurements locally on performance logs [5]. These measures are usually processed locally for monitoring purposes or stored and processed for posterior analysis.

III. ISO 25000 SOFTWARE QUALITY ASSESSMENT STANDARD

A. 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).

B. 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.

C. 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" and a derived measure is a measure that describes a function of two or more values of base measures [19], and are derived respectively from a **measurement method and a measurement function**. Identical definition is proposed in the International Vocabulary of Basic and General Terms in Metrology [20]. The quality measure elements (QME) are either a base or a derived measure, which means that a LLDM could be a QME. [21, 22]. This definition is an adaptation of the one in the International Vocabulary of Basic and General Terms in Metrology [4] [23]. 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 [23]. It is also based on the definition in [20]. A measurement function is defined as an algorithm or calculation that combines measures. [25]

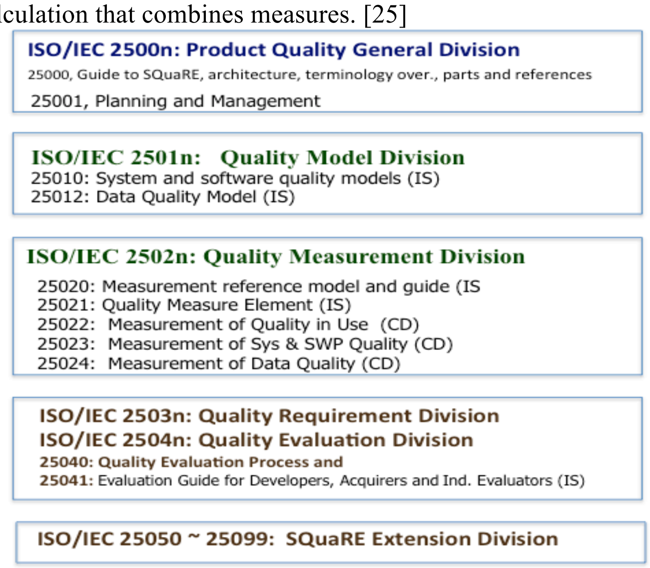

Figure 1: Five document groupings

IV. CLOUD COMPUTING

As we have presented in the introduction, cloud computing is a complex technology that depend on different infrastructures that include components that are often

[3] SNMP is a "Internet-standard protocol for managing devices on the first approach offers on IP networks".

[4] 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.).

dispersed geographically, with shared elements and running diverse applications [26]. This technology employs hardware and software to deliver ubiquitous, resilient, scalable, billed-by-use, application agnostic systems [12]. In the scope of this research, the cloud-computing infrastructure analyzed fits the classification of a Private cloud.

One of the frequently cited sources for the definition of cloud computing is the US National Institute of Standards and Technology (NIST), that proposes that "*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*" [27].

Cloud computing is offered or assembled in different formats to the consumers. Three formats are the most prominent: Infrastructure as a Service (Iaas), Platform as a Service (PaaS) and Software as a Service (SaaS).These services can be deployed in different formats, mostly constraining cost, administrative effort, customization and privacy requirements, being Public, Private and Hybrid.

Infrastructure as a Service (IaaS) is an offer where a provider offers virtual or physical computing resources (CPUs, memory, disk space) over which a customer is free to deploy and manage its own environment. This allows a greater degree of customization, but causes a larger overhead in management processes to the client. Amazon Elastic Compute Cloud is one example of such a service.

Platform as a Service (Paas) is a different offer whereas a set of computing resources, operational systems and development tools are hosted by the provider and the customer is capable of creating services and applications that are compliant to the offer's characteristics, with a limited degree of customizability. This offers greater stability and control of computational resources, as the customer can focus on developing or hosting the products and services owned without having to spend resources in managing, updating and maintaining the infrastructure. One such offering is the Windows Azure Platform.

Software as a Service (SaaS) is a form of offer where the consumer accesses applications, services and information from a standard interface, having low customizability but no administrative effort. These applications are hosted and managed completely by the provider. One such application is the widely used Gmail application by Google.

Public Clouds are owned, managed, configured and controlled by the service providers who can then offer the cloud third party clients. **Private clouds are built for specific organizations**, with the possibility of outsourcing its management to third parties. **Hybrid clouds** contain one or more components that are owned by private and public parties.

SaaS Description of this case study: In this case, the evaluated SaaS is responsible for servicing e-mail clients, encompassing the desktop application, active directory authentication, network transport, message storage and indexing. The minimum system requirements are described on [28].

V. CASE STUDY

In order to address the research problem of the possibility of employing data center logs to model degraded cloud computing application performance via the monitoring of end user's reports of such degradations, the authors perform an exploratory case study where users complaints, in the form of incidents or trouble tickets, reported to a help desk, are studied during an specific work period. Whenever these trouble tickets relate to the studied SaaS application, the performance logs from the all the nodes represented on figure 2 are collected in a performance management database. These performance measures undergo the 3 steps presented, at the end of section I.

The following methodological protocol is applied during the case study:

Data Collection: Data is collected from two different sources: a) the Information technology Service Management system (ITSM) that is accessed and maintained by the help desk for record keeping and b) the data center logs collected. During the case study we received 30 complaints at the help desk and collected approximately 4 GB of datacenter logs for this application.

Data organization: For the help desk tickets, data is concentrated on the smallest time segment possible in order to represent the most amount of complains with as minimal environmental variation as possible. For the performance logs, three different work windows are open: 1) the moment of the degradation report at the help desk, 2) the previous three hours, and 3) the anterior week. After the data collection data is collected, the LLDM are associated to the ISO quality characteristics. Then we conduct data analysis. Two distinct processes are used for analysing the data. First the ticket information is manually read to identify clearly the performance issues. Second, statistic data of the logs is

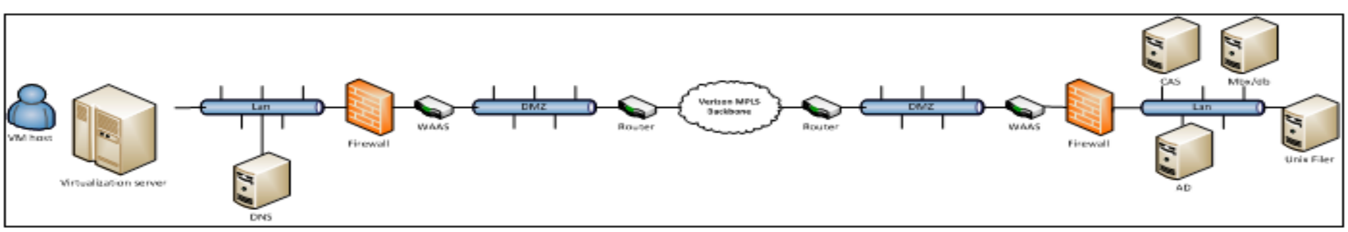

Figure 2: A Private SaaS cloud that is used on this research

compared for the three previous work windows as well as in between reports in order to identify similarities between the degradation reports. On this research, the logs employed follow a "RAW" format, whereas the performance data is represented by "Timestamp: LLDM name; LLDM Value". There is no metadata available on these logs that can help on accelerating indexing or searching the information.

Data interpretation is conducted in order to identify the possibility to map the user complaints to the LLDM's and then the LLDM's to the ISO quality characteristics, which would offer a method for monitoring LLDM's to 1) understand the user perspective; and, 2) generate quality indicators for the application under study.

Once the main measurement steps have been done, the following data processing is done.

Collection of user degradation reports: This involves monitoring an Information Technology Service Management (ITSM) system for new tickets, searching for keywords such as "e-mail", "slow", "slowness", and "hanging". For each complaint that matches the keywords, the machine name and time stamp are recorded. During this case study, 30 such cases were observed, covering 45 minutes.

Logs collection. For the 30 above cases, the relative data available has been collected for further analysis. This amounts to 4GB of data organized initially in different files, then transformed in a NO-SQL file for statistical analysis. The NOSQL files are organized as the lines represent the time series and each column is a different LLDM

Association of the LLDM. This is an adaptation of quality characteristics of ISO 25023 standard.

Statistic exploration of the low level derived measures. The LLDM identified during step 2 are then compared, using covariance and correlation techniques, aiming to reduce the total amount of observed data. Then, skewness and kurtosis of the 3 work windows are calculated, in order to establish a baseline (week), and escalading scenario (for the three previous hours) and a reported event (the hour). This allows us to see if there is any difference between the baseline and the actual degradation report. Furthermore, principal component analysis (PCA) is calculated to determinate measures with the most impact. From the PCA, frequency and trend lines are determined for the values, in order to link the values back to the user reports of degradation (i.e. the Help desk tickets). This helps in identifying a) which measures has more signification and b) to which extent they affect the user experience. This step is ongoing as of the writing of this report.

A. Current results:

In this section we present the results of each of the methodological steps and sub steps presented on section 5[1-4], beginning by the presentation of the step-by-step approach of the execution of this case study:

1 – <u>Identify degradation report</u>: The tickets logged at the help desk are analyzed for the keywords (i.e. "slow", "hanging", and "slowness", amongst others). Tickets, which contain these keywords, are flagged as potential performance degradation issues.

2 – <u>Data extraction and organization process</u>: Extract the raw performance data associated with the performance degradation report, for 1 week of time. 63589 data points were collected, with 38 LLDM. We observe 33 high degree correlations (i..e. >+0.74), while 12 presented a strong negative correlation (i.e. <-0.60), from which we reduced the 38 initial measures to 15. These have been selected based on the described statistics approaches and also based on logical response of being regarded as being available when the value is lower; for example, Memory_Commited_Bytes is selected instead of Memory_Available_Bytes, mainly because both are strongly uncorrelated (i.e. -0.98) and because the smaller amount of committed bytes, more will be available. For this case study, the selected list of LLDM, named as per the according logs, is: %_Processor Time; Page_File_%_Used; Commited_Bytes; AVG_Disk_Read_Queue; I/O Read; Private_Bytes; Thread_Count; Handle_Count; AVG_Disk_Write_Queue; Connection_Failures; Pages/Sec; Connections_Active; Connections_Reset; Disk_Free_MB;

These LLDM are then associated to the Performance Measurement Framework Cloud Computing Concepts proposed by Bautista, resulting in the classification demonstrated on table 1. For the indicators demonstrated on this paper, only Performance measures have been utilized as further described on section VI.

Using this data organization, it is possible to plot the graphical representation of the ISO 25023 quality concepts as demonstrated on figures 3 and 4.

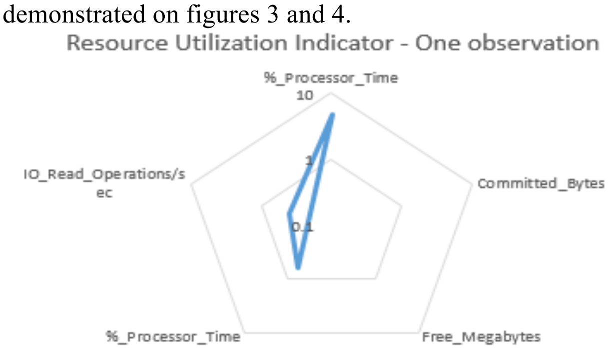

Figure 3 – Single observation of the resource utilization values

Figure 3 represents one single data point in time, whereas figure 4 represents the collection of all data points for one specific concept – Performance Efficiency: Resource Utilization.

Table 1 – ASSOCIATION of LLDM and ISO 25023 CONCEPTS

Name of the Selected LLDM as extracted from the logs	**Concept to which the LLDM can be associated according to ISO 25023**
%_Processor Time Commited_Bytes Disk_Free_MB	Performance Efficiency – Resource Utilization

Process_%Processor_Utili I/O_Read Private_Bytes	
Page_File_%_Used Avg_Disk_Read_Queue Avg_Disk_Write_Queue Connections_Active Pages/Sec Handle_Count Thread_Count	Performance Efficiency – Capacity
Connections_Failures	Reliability – Maturity
Connections_Reset	Reliability – Fault Tolerance

Figure 4 is purposely left on this format to represent the challenge of interpreting all performance data for the cloud simultaneously. This is then comparable to the diagram present on Figure 5.

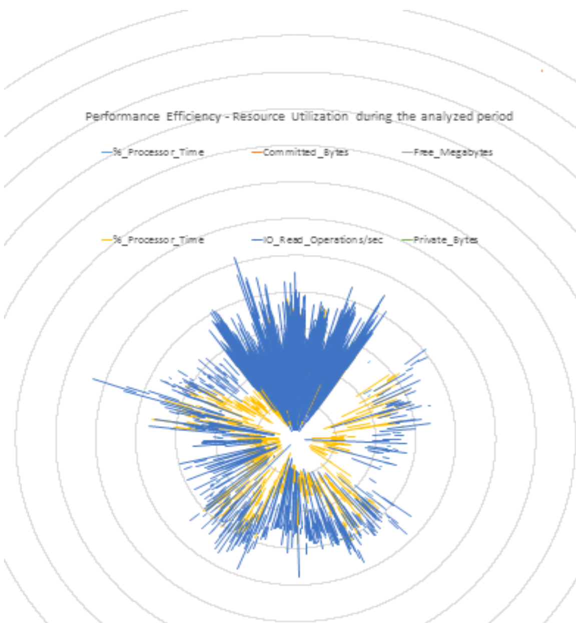

Figure 4 – Multiple observations of the resource utilization values

Figure 5 represents the resource utilization indicator, demonstrating visible peaks on the observations 32, 5226, 11132, 22651, and 37022.

These observations correspond to the following time stamps: Monday 07:56 AM, Tuesday, 08:05 AM, Wednesday 10:28 AM, Thursday 07:53 AM and Friday 10:38 am. These peaks in resource utilization could point towards a pattern on resource utilization on those specific times of the day, possibly indicating a degraded performance from the perspective of this quality characteristic.

Upon reviewing 30 of such reports along with the steps that were taken for recovery, in 24 cases the technician giving the support observed either or a combination of 4 LLDM - processor, memory, disk space and network utilization – to determine the possible root cause for each case. In at least 6 of the cases, the user who first presented the complaint reported a recurrence of the issue.

Eighty-four different LLDM exist on the logs relative to these 30 cases. Twenty-two are strongly correlated (+0.74) and 19 are strongly un-related. If the empirical model of 4 measures seems simplistic, the exploratory model of 84 measures can be optimized via correlation and covariance, for reduced measures. Utilizing population variance, 14 LLMD demonstrate higher comparative significance

For these LLMDs, the skeweness and kurtosis is calculated, generating a possible classification into generalizable (low kurtosis) and non-generalizable (high kurtosis)

By observing the values of the LLDM, it is possible to link positively the higher values with the complaints of the users, indicating that the empirical knowledge disclaiming that the lowest levels of utilization will provide better user experience. With the association of the measures in quality characteristics, it is possible to fundament the creation of quality indicators derived from LLDM that can represent the user perception of such derivations, possibly leading to an indicator of the service level of the cloud computing system.

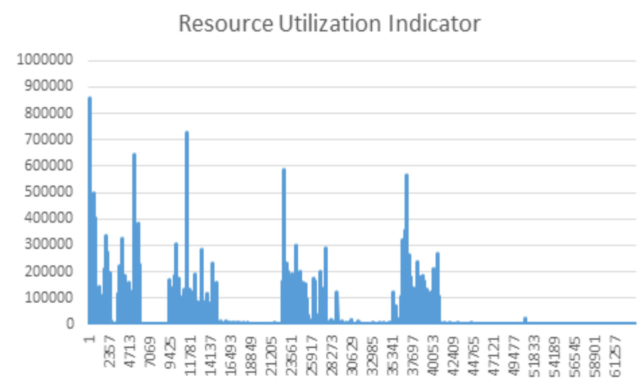

Figure 5 – Resource Utilization Indicator

Figure 3 represents the collection of a single data point of the resource utilization for this application. Figure 4 illustrates the complexity of demonstrating all the data points in a single graph, which prompts the creation of the resource utilization indicator demonstrated on Figure 5.

VI. Challenges and Conclusion

During the planning of this case study, there was some expectation that the experimentation would be laborious because of the many manual processes involved with implementing the ISO quality standards concepts as well with the many statistics analysis involved:

-Data collection challenges: Reading through help desk tickets might not be the best method for extracting user reports of performance degradation. Issues about misspellings, synonyms, different technician interpretation all combined created, initially, some subjectivity. Additionally, extracting and transforming the data from the existing log tools format, which is stored in a relational database, to convert to a NO-SQL database format (i.e. in order to be able to perform the statistical analysis) was also a difficult process initially; whereas the data is exported form one system in text format, then a NO-SQL table has to be designed and the data imported.

-Data analysis challenges: the recursive calculations necessary for the PCA and outlier detection require computing power. Performing these calculations, in real time, have proven to be a challenge.

-Data interpretation challenges: even though the statistical techniques help for reducing the amount of data giving quantitative data for decision making, there are still subjectivity degrees involved in analyzing the data; the analyst's ability to interpret the data, especially in comparison to other data sets, may influence the results of the data analyses.

Despite these challenges, it was possible to a) manually associate the LLDM as per Bautista's framework, which expands the original author's work to include real world data; b) statistically analyze the data in order to produce tentative indicators, which promote a better understanding of the performance of the application as the interaction of all its composing parts. The method, the framework and the processes are still far from conclusive, as expectable for a research in progress.

VII. Next steps

As described, this short paper presents the results of a research still in progress. The following activities are underway:

-Automated data extraction, consolidation;

-Automation of the baselines, correlation and co-variation calculations;

-Automation of the data reduction;

-Frequency analysis for outlier detection of the performance values, which would strongly link the values with the end user perception of performance;

-Implementation of quality indicators

References

[1] J. C. Emery, "The Impact of Information Technology on Organizations," 24th Annual Meeting, Academy of Management, p. 1, 29 December 1964.

[2] R. Buyya, C. Yeo, S. Venugopal, J. Brober and I. Brandic, Vision, hype, and reality for delivering computing as the 5th utility., FGCS, Volume 25, I 6, June 2009, 599-616., 2009.

[3] S. &. W. S. Davis, " The mediating effects of intrinsic motivation, ease of use and usefulness perceptions on performance in first-time and subsequent computer users," Interacting with Computers, 13(5), pp. 549-580, 2001.

[4] J. F. A. Etezadi-Amoli, A structural model of end user computing satisfaction and user performance, ELSEVIER, Information & Management 30 - 65-73, 1996.

[5] Microsoft, "Microsoft Perfmon," 2013. [Online]. Available: http://technet.microsoft.com/en-us/library/cc749249.aspx. [Accessed 05/ 2014].

[6] Nagios, "Nagios IT Infrastructure Monitoring," 2013. [Online]. Available: www.nagios.org.

[7] J. Weisberg, "Argus TCP Monitor," 2013. [Online]. Available: http://argus.tcp4me.com. [Accessed 05/ 2014].

[8] Microsoft, "Performance Analysis of Logs Tool," 2012. [Online]. Available: http://pal.codeplex.com/.[Accessed 05/ 2014].

[9] A. Friedl and S. Ubik, Perfmon and Servmon: Monitoring Operational Status and Resources of Distributed Computing Systems, CESNET Technical report 1/2008, 2008.

[10] R. Kufrin, Measuring and improving application performance with Perfsuite, http://perfsuite.ncsa.illinois.edu/publications/LJ135/, 2005. [Accessed 05/ 2014].

[11] 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. [Accessed 05/ 2014].

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

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

[14] 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.

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

[16] L. Bautista, A. Abran et 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.

[17] Omniti, "Reconnoiter Fault Detection and Trending," 2013. [Online]. Available: https://labs.omniti.com/labs/reconnoiter. [Accessed 05/ 2014].

[18] Cacti, "Cacti RRDTool," 2013. [Online]. Available: http://www.cacti.net. [Accessed 05/ 2014].

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

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

[21] ISO/IEC, 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.

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

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

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

[25] A. ALinezhad, A. Masaeli and N. M. M. Esfandiari, "Evaluation of Effectiveness of Implementing Quality Management System (ISO9001:2000) Using BSC Approach in NIGC", Journal of Industrial Engineering 6, 33-42, 2010.

[26] N. Mirzaei, cloud computing, Institute Report, Community Grids Lab, Indiana.Edu, 2008.

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

[28] Microsoft – Exchange 2010 system requirements – Available on https://technet.microsoft.com/en-us/library/aa996719(EXCHG.141).aspx [Accessed 12/2014]