Comparative Study of Scalability and Availability in Cloud and Utility Computing

[1] Farrukh Shahzad Ahmed, [2] Ammad Aslam, [3] Shahbaz Ahmed, [4] M. Abdul Qadoos Bilal

[1] Netsolace Information Technology PVT (LTD), Islamabad
[2] White Wings PVT (LTD), Islamabad
[3,4] Department of Computer Science & software engineering , International Islamic University Islamabad
{[1] fshahzad11@gmail.com, [2] amaa08@student.bth.se, [3] shahbaz.ahmed@iiu.edu.pk}

ABSTRACT

Cloud computing and Utility computing paradigms are two resource sharing architectures. The vivid and multi-institutional natures of these environments instigate different challenges in context of availability and scalability. In this report we discuss the normal architecture of Cloud and Utility computing followed by crucial areas which are availability and scalability. To address these problems we proposed a new controlling and scheduling mechanism, Optimize Scheduler Authenticator and Controller (OSAC) [1]. Qualitative and quantitative research strategies are used to emphasise on these areas. Experiment is conducted to get a comparative view of availability in a real context. Interviews are used as mean of data collection for scalability issues.

Keywords: *Scalability, availability, cloud computing, utility computing.*

1. INTRODUCTION

Extensive study of both paradigms revealed that they rely on same underlying infrastructure inherited from Grid computing infrastructure. One can be differentiated from other on the basis of its implementation. In attempt to address to address scalability and availability in these two paradigms, we proposed Optimized Scheduler Authenticator and Controller (OSAC). General overview of these areas is addressed in this paper while low level implementation details of OSAC and communication mechanisms between different OSAC's will be addressed in future.

Ever increasing use and popularity of Internet impel Internet into a distributed computing platform. This shift has been persuading companies worldwide to outsource their computing resources, business processes, business applications and data storage and maintenance to get the benefit of up-to-date IT technologies in order to focus on their core business competencies [2] to survive and compete. This survival competition is the key driving factor for evolving the Internet into a distributed computing platform. Diverse support, ability to scale from small networks with few devices to many devices up to a global scale and support for wireless technology is some of intriguing features of distributed networks for companies. The support for the new devices will increase in future [3]. Cloud and Utility computing envisioned as next generation computing platforms [4][5]. Extend traditional distributed computing providing large scale sharing of storage and computation resources [1]. Grid computing is defined as, “A system that uses open, general purpose protocols to federate distributed resources and to deliver better-than-best-effort qualities of service” [6]. Utility computing is defined as, “A collection of technologies and business practices that enables computing to be delivered seamlessly and reliably across multiple computers” [7]. The idea of a Cloud is a system which has loose boundaries and can be able to interact and merge with other such systems. There is no precise and comprehensive definition of cloud computing yet available. The notion is that the applications run somewhere on the Cloud which users are least concerned about. The whims of both paradigms are still overlapping. Cloud computing relates to underlying architecture in which services are designed which may perhaps equally apply to Utility services [8]. Couple of definitions of Cloud computing are: According to Gartner Cloud computing is, “A style of computing where massively scalable IT-related capabilities are provided „as a service‟ across the Internet to multiple external customers” [9]. According to Aaron Weiss Clouds computing is: “Powerful services and applications are being integrated and packaged on the Web in what the industry now calls “cloud computing”” [10].

As mentioned earlier the Cloud and Utility computing are comparatively new and evolving areas in IT industry and there is lot to be done. The foremost concerning issues are Scalability and Availability. The purpose of this paper is to study the following issues in Cloud and Utility computing paradigms,

1. Availability

2. Scalability

2. BACKGROUND AND RELATED WORK

Cloud computing addresses both platform and application [11] whereas Utility computing is the combination of computing resources as a metered service. It is a service level agreement SLA between the user and the service provider like any other physical public utility [5]. Underlying architecture differentiates both computing architectures from each other. Both are Service Oriented Architectures (SOA) where services (combination of hardware and/or software) are delivered on demand. Utility computing delivers application infrastructure resources [8] correlation to business. On contrary, in Cloud computing applications are developed and deployed in a way that they can run in virtualized environment. Dynamically allocating and sharing of resources which allow them to grow, shrink and self heal. Cloud computing is characterised by this dynamic behaviour. James Governor, the Analyst for RedMonk, has been an IBM and Microsoft corporate watcher for 8 years argued that machines in Cloud architecture are not visible [12] to the users. These resource pools in Cloud architecture can be located anywhere in the world [13]. Contrary to Utility computing, which offers Software as a Service SaaS (examples are Salsesforce, Gmail, Gliffy), Cloud computing also offers Platform as a Service PaaS (examples are Mosso, Google App Engine, Rails One) [1] applications run on virtual operating system. Another innovation in the Cloud computing which differentiate it from Utility computing is Infrastructure as a Service (IaaS). Companies like Joyent, Amazon Web Services, Nirvanix etc., are offering whole computing Infrastructure as a Service from online development platform to normal computing requirements. Systems based on the Cloud computing framework can interact with each other, sharing and pooling resources for greater efficiency over a large deployment such as an enterprise [14]. Complex multi-tier nature of enterprise applications makes it challenging to deploy on Cloud framework. As a result, dynamically adjusting resources to an application not only has to take into account the local resource demands at node where a component of that application is hosted, but also the resource demands of all the other application related components on other nodes [8].

Related Work

To the best of our knowledge, no experimental studies have been conducting that compare and evaluate the availability of the resources between two different architectures. Further no comparison has been made between two computing service providers in terms of scalability.

Optimized Scheduler Authenticator and Controller (OSAC)

In enterprise applications, different amount of resources are required at different tiers [15]. Despite the fact that resources are available, data centres are often not fully utilized due to this vary reason. The role of OSAC is to allocate resources intelligently in a way that they are fully utilized, considering the performance and requirements parameters. A generic architecture of OSAC is given in figure 1 [1].

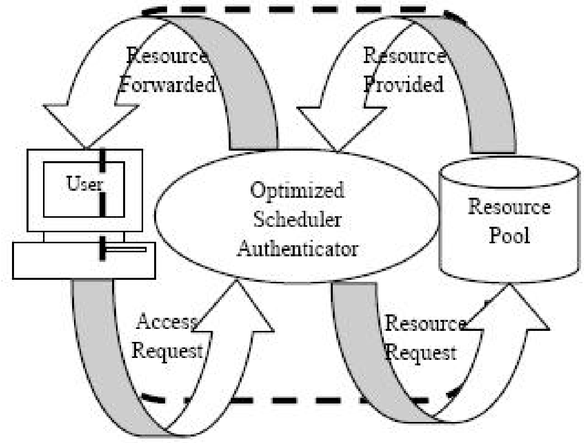

Figure 1: Generic Architecture of Optimized Scheduler Authenticator and Controller

Working of OSAC

Optimized Scheduler Authenticator and Controller OSAC is comprised of two modules, first one is the Optimized Scheduler and Authenticator OSA and second is termed as Optimized Controller OC. OSAC runs on virtual network layer, the core of Cloud and Utility computing infrastructures. Users connect the network via OSA which besides authenticating user also schedules and allocates resource availability to the user and its processes. In other words OSA schedules user's resource requests. On further level down, OSA has further two sub-modules the Optimized Scheduler and the Authenticator. The user is authenticated by Authenticator while Optimized Scheduler schedules the resources for the authenticated users. The token request for the required resource is passed to OC which in turn provides the available resources list to OSA. OSA then intelligently on the basis of best possible performance allocates the requested resource to the user. The parameters for calculating performance are latency, network traffic and bandwidth required by the user.

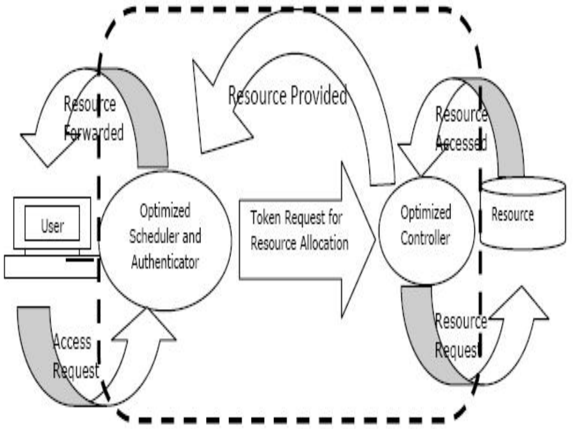

Figure 2: Architecture of Optimized Scheduler and Controller

When ever any resource becomes unavailable or any node goes down, OSA without interrupting the user checks the availability of the resource in the resource pool through OC. OC provides the current updated list of the available resources to the OSA which will in-turn again allocate the resource to the user. As soon as user finished up with the resource and resource is free, the OSA returns the resource to the resource pool by notifying to OC. The interesting feature of OSAC is that as soon as OC gets an update about any free resource from OSA's it will notify to OSA's the updated list of resource pool which they utilize to recalculate the allocation of that resource for the users currently using the specific resource. The resource is then reallocated to the user without interrupting its processing.

Availability

Availability of system or component is fraction of time it is available and it describes the system behaviour. It is defined as, "The degree to which a system or component is operational and accessible when required for use" [16]. Availability is calculated as, Availability = Uptime / Uptime + DownTime = MTBF / MTBF + MTTR Mean time to recover / repair (MTTR) - Average time it takes to recover Mean time between failures (MTBF) - Average time between failures

If MTBF is much greater than MTTR then, Availability ≈ 1 – MTTR / MTBF

The system with 0.99 availability has 1- 0.99 = 0.1 probability of failures. Availability measures give the reliability of system which is defined as:

"Reliability of a system is the probability, over a given period of time that the system will correctly deliver the services as expected by the user" [16].

Table 1: Uptime and Maximum Downtime

Uptime	Uptime	Downtime per Year	Downti-me per Week
Seven nines	99.99999%	0.3 s	6 ms
Six nines	99.9999%	31.5 s	0.605 s
Five nines	99.999%	5 min 35 s	6.05 s
Four nines	99.99%	52 min 33 s	1.01 min
Three nines	99.9%	8 hrs 46 min	10.1 min
Two nines	99.0%	87 hrs 36 min	1.68 hrs
One nine	90.0%	36 days 12 hrs	16.8 hrs

Availability, reliability and performance are though different terminologies yet they are closely link to each other. Goal is always to achieve the best throughput from the resources provided they are reliable and available when needed. The nines of availability (shown in table 1) best describe in which category the system lies. Achieving high availability is always expensive. But for the critical systems, like life airplane system computers, defence systems seven nines availability is required whereas telecom, navigation, banking and ATM‟s can rely on six nines. Office systems and messaging systems three and four nines availability respectively serve the purpose.

New enterprise data centres are now being planned with a Cloud and Utility computing architectures. All hardware resources and in some implementations applications as well (for example Google App Engine) are pooled into a common shared infrastructure and users share these resources on demand basis which change over time [17]. The request for the resources is scheduled by OSA while controlled by OC as explained above.

3. EXPERIMENT PLANNING

We have conducted an experiment to find and compare the availability of Amazon Elastic Compute Cloud (EC2) architecture with OSAC architecture. As the experiment is not real time so we have made following assumptions [1]:

1. Normal and controlled working environment.
2. The values are not real and are supposed for the experiment.
3. No extra functionality is added into the system.
4. The systems that are requesting for resource allocation are from same level of performance.
5. All the systems are requesting for the same type of resources.
6. All systems are in operational mode from the same time.

Hypothesis Testing

In our experiment we have used both Null Hypothesis and Alternative Hypothesis defined as follows,

H0: Probability of failure in OSAC architecture will be higher than Amazon EC2 architecture. **H1:** Availability of resources in OSAC architecture will be lower as compared to Amazon EC2 architecture.

Variables Selection

Both dependent and independent variables are used for experiment. Detail is as follows:

Independent Variables

Mean Time to Recover (MTTR):

Mean Time to Recover determines the average time taken by the device from recovering any type or failure.

Uptime:

Uptime defines for how long the system is running or "up" for example, 10 days, 30 days etc.,

Downtime:

It determines for how long system's resource(s) is not available for access.

Dependent Variables

Mean Time between Failure (MTTB):

It is used for measuring the failure average time that occur in between the system. It is calculated as:

MTTB = (Number of system x time period) / number of failure during that time.

Availability Variables:

It is depended of the Uptime and down time and defines that how often the resources are available.

Instrumentation

To conduct the experiment, the required resources are prepared and installed on machines on which the test is to be performed.

- All systems are connected to high speed broadband internet connection.
- The Amazon EC2 and OSAC service available on all systems

For measuring the independent variables, system clock software is installed into the systems. It is used to gather statistical data for the time for failure, uptime and downtime. Besides independent variables, the dependent variables are also measured with the help of software to get whole picture.

Experiment Design

The experiment is based upon three types of design.

Randomized Design: As this experiment is based upon randomized design so, each system has been given access to make a request for resources to any type of service whether from OSAC or Amazon EC2. A system that is using a service of one the architectures may ask for a service to other architecture, for example a system that is using Amazon (EC2) service may request to utilize the service of OSAC.

Balancing: To ensure balance in experiment equal types of resources accessed by the systems (objects) from the both services. The broadband connection is same and the systems that are used have same specifications.

Blocking: As the experiment is taking place in an open environment i.e. through internet therefore, every computer can access resources from these services. An ID is allotted to each system that is verified before accessing the resources and the systems that do not contain any type of ID will be blocked ultimately.

Experiment Operation

Preparation

Data is gathered through automated software. This minimized the user physical interaction and human errors resulting in precise results. Software use different algorithms for randomly accessing the resources and different services.

Execution

The total length of the experiments was 15 days and 6 hours. All the systems access the services as it should and no system result in failure.

The average time periods and results are calculated from special software installed on each system. It also validated from the log files on the servers that all the systems have access the resources that make sure that balancing, randomization and blocking took place in an intended way.

Validity Evaluation

Like other scientific experiments our experiment has also some types of risks associated with its valid ness. This section is covers these risks. The validity of this experiment is based on validity evaluation framework proposed by Wohlin [18].

Internal Validity

It is the approximate accuracy about inferences concerning cause-effect or causal relationships [19]. The experiment is conducted on number of computer systems constitute a group. Therefore, posing single group risks rather than multiple group risks.

History Risk

All the systems have the same level of specification. They all belong to the same level of effectiveness and history so there is no risk involve in this experiment related to the selection history.

Maturation Risk:

This risk concerns our study on account of different requests from the system for different resources.

Testing Risk:

It is not valid risk for our study because we are concerned with how pre-test conditions related to the post-tests.

Instrumentation Risks:

As we are using different types of software and services so, chances are there that produce inaccurate results. This is major level of risk into our experiment and all the outcomes may become false due to this.

Mortality Risk:

All the systems are from the same level of functionality and devices or software installed on it. Therefore, it is again not concern with this specific experiment.

Regression Risk:

It is also not a valid risk because machines are participating here and the selection is not based upon there pervious functionality.

External Validity

It is the degree to which the conclusions in the experiment would hold for other persons in other places and at other times [20]. In our experiment we are using the systems as objects. The specification are the same, but in general term we are unaware of the other systems working with different specifications, bandwidth and in work environment.

The other risk concerns with limited scope of our test. We are comparing OSAC only with Amazon (EC2), there are many other architecture available in the market that can produce better results with our proposed architecture.

Another risk is in systems softwares' that are installed. These softwares direct as to which operation to be performed but generally in actual practice with human interference, our results may or may not hold their validity.

Construct Validity

Interaction of Different Treatments:

This thread is involved in our experiment because the systems used in the experiments many associate in other type of performing some other types of functions at the involved concurrently in some other programs designed to have similar effects at the same time.

Restricted generalizability across contracts:

Although the same software are installed on all machines but there is a possibility that one software don not perform a better resource allocation requests form the services and end-up with false construct validity.

Confounding Constructs and Levels of Constructs:

It is thread that on the sequence of systems requests the OSAC is performing better while compared to Amazon (EC2. But, when used in industry where the requests sequence is different than this architecture may not perform better.

Conclusion Validity

Low Statistical Power:

This threat is present because some systems may not take part in the experiments i.e. system error and hardware failures.

Reliability of Measures:

We are measuring the availability with the specific and standard formula so there is risk involved in terms of the reliability of the outcomes.

Data Analysis

It is used to illustrate the fundamental features of the data in an experiment. With the help of graphic analysis they provide simple summaries about the sample and the measures [21]. The procedures used for data analysis propped by Wohlin [18], based on the following steps, Descriptive statistics Data set reduction Hypothesis testing

Descriptive Statistics:

It is used to illustrate the fundamental features of the data in an experiment.

In table 1, we apply how much the availability a system can provide using the current architecture of the Utility and Cloud computing. We test the system with different set of values and find that the probability of failure is very low when it provides higher availability but in most of the cases its values do not satisfy the required availability failure result.

Table 2: System availability Analysis under current structure

Availability	Probability Failure
0,99	0,01
0,89	0,11
0,91	0,09
0,92	0,8
0,88	0,12
99	0,01
95	0,05
90,2	0,011
93,5	0,65
93,8	0,62
95,1	0,049

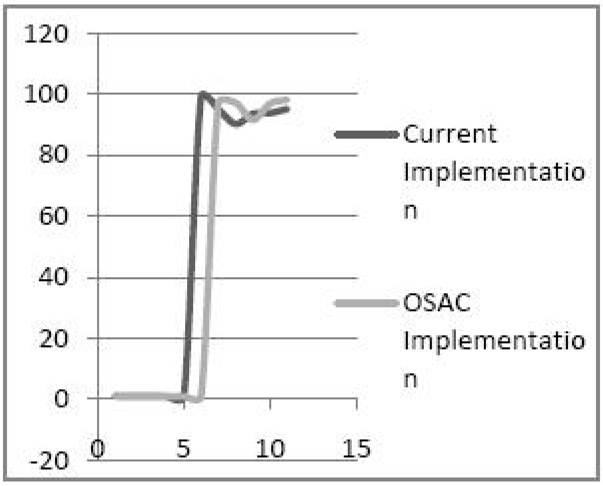

Figure 2: System Availability Analysis of Current Structure (EC2)

Data Set Reduction

When we compare this architecture with our proposed OSAC by applying the same set of values it is found that there are comparatively less occurrences failure. After taking the data from description statistics approach, researcher use data set reduction to present the data. This is due to the outliners presented in the dataset that may influence the results [18]. To produce better results of the experiment we identified an outliner in our dataset that is having unusual behaviour from the dataset. A value that is unusual is excluded due to the power failure of a system with the help of scatter plot. If we include this value it changes the whole scenario of the results.

Hypothesis Testing

We have performed Hypothesis testing on the null hypothesis that is, H0: Probability of failure in OSAC architecture will be higher than Amazon EC2 architecture. The null hypothesis testing is based upon the samples taken from statistical distribution. A sample is selected to reject the fact of null hypothesis [18]. The factor used here is availability and as discussed above, the experiment type is single group, single treatment. Therefore, we used Chi-2 method because it deals with frequencies of data [18]. Table 2 defines the frequencies as they are obtained during the analysis of statistical distribution. We have used ordinal measurement scale because this method is selected as non parametric test based on design type. The results of the experiment are collected by executing the test in order to draw the conclusion.

4. DISCUSSION

After executing the experiment and gathering the results it is analyzed that the OSAC architecture is a better architecture for the availability of the resources. The results nullify the null hypothesis and support the alternative hypothesis. The result shows that the current availability does not provide the higher availability. Figure 7.3 defines that the current system is not appropriate to use and the chances are more towards the unavailability of the resources. OSAC provide a comparatively more system availability

Table 3: System availability Analysis of OSAC

Availability	Probability Failure
0.99	0.1
0.95	0.05
0.96	0.04
0.92	0.08
0.93	0.07
0.963	0.37
96.8	0.032
97.1	0.29
91.5	0.085
96.8	0.032
98.1	0.019

Scalability

The notion of scalability is that “How well the solution to some problem will work when the size of the problem increases” [22]. There are many different vendors for example Hewlett-Packard, Sun Microsystems, IBM etc., are offering Cloud and Utility computing services. Scalability has become an important aspect of the infrastructure. Different vendors are providing different types of services on Cloud and Utility computing and they have the different level of scalability along with different definition of it as there are no definition available that defines the scalability in a universal term. The system is called un-scalable if the additional cost of coping with a given increase in traffic or size is excessive, or that it cannot cope at this increased level at all [23].

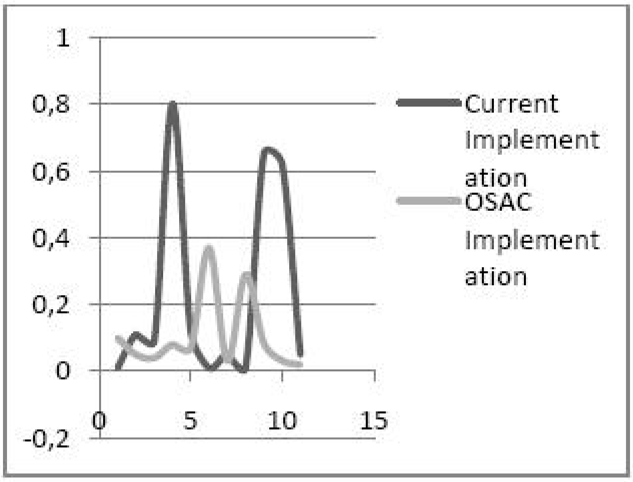

Figure 3: Probability of Failure in Current and OSAC Architecture

Qualitative Research Strategy

The qualitative research is devise to start with data collection which will be carried out interviews and observation. There are two different types of action performed in data collection process. One is data elicitation and other is data recording for interpretation of data in the textual form that would be used afterward for data analysis.

Data Collection

As there are many different companies offering these services so we have chosen two different companies IBM and Hewlett-Packard (HP). Data collection is the done by taking the semi-structured interview from the users (objects) and observations are made on the basis on interviews.

Interviews are used for data collection because it is a simple, reliable and powerful approach to get accurate and precise information from the customers of the service. In structured part of the interview the questions are predefined in for scalability testing. The interviewers are free if they like to mention any other issue that is not available in the interview question list or any other problem that is not related to scalability.

IThe interviews are conducted from the users that have an experience about both companies in terms of scalability. The interview is a video-recorded so that host does not waste his time in taking the notes from what a customer says. Despite that they should spend more time for taking the accurate answers from the object. As scalability concerns much about marketing so, the questions in the interview that are mostly related to marketing managers are prioritizing as compare to other stakeholders like developers or designers etc.

Data Analysis Procedure

Data is gathered through videos. The video is then converted into textual form. The text not only contains the speeches during the interview but also the expression such say voice expression, feelings, and expression language. All expressions are taken so that we can analyze the data in an accurate way and to produce accurate research results. The results are also distributed into a presentable format to the different stakeholders of the company i.e. designers, developers so that they can analyze what they are required from the computing service. The analysis produces the results that IBM computing services are providing better scalable services to the user as compared to HP.

Validity of the Study

A possible risk may be lead to misunderstanding of the subject while answering a question. For example, Cloud computing and Utility computing is based upon the Grid computing and a person might answer the question in the context of grid computing. The solution to this threat is provided in this way; the host will provide an example to the asked question so that the subject do not confused and answer the questions in the required manner. The other threat is from worthless data. We are analyzing all the data in the textual form. This data will also involve raw data which is useless in the qualitative research; the involvement of this worthless data in the research may lead us towards inaccurate results.

Expected Outcome

By implementing IT infrastructure on Cloud or Utility computing paradigms, company will gain increase in its revenue as both paradigm are scalable to greater extend are offering to accommodate infinite number of users. The major expected outcome from the study is to find out which company is providing better computing service in terms of scalability. As discussed above, scalability is used mainly as an advertisement source. Therefore, the main stakeholders in out research are the marketing managers.

The results are provided to the market departments of both companies so that they can analyze which what are the strengths and weakness of them. It also helps them to deign the strategy of their business both in terms of internal and external appraisals.

5. CONCLUSION

Using shared resources is a cost effective approach. Cloud and Utility computing provides a infrastructure where user can use these shared sources. Both paradigms offer services on demand. Resources are allocated, de-allocated, configure and reconfigure dynamically within pre-defined rules by the service provider on contrary to the Utility computing which denotes both a separation between service provider and consumer with the provision of having desired set of rules defined by the user. After comparing the two aspects of the two paradigms Scalability and Availability, we proposed our own controller structure OSAC. OSAC is a generic controller and aim to produce better throughput then the current implementation. The experiment proved that the availability of resources is better allocated by the OSAC architecture. There is a big room for the future researchers to address these three issues fully as well as other concerning issues.

REFERENCES

[1]. A. A. Nauman, A. Aslam, B. Garapati, "*Cloud Computing Versus Utility Computing: A Comparative Study of Availability, Scalability and Security Aspects of the Two Paradigms*", Department of Computer Science *Blekinge Institute of Technology, Ronneby, Sweden.*

[2]. M. J. Buco, R. N. Chang, L. Z. Luan, C. Ward, J. L. Wolf, P. S. Yu, *"Utility computing SLA management based upon business objectives"*, IBM Systems Journal, vol. 43, no. 1, 2004.

[3]. M. Milenkovic, S. H. Robinson, R. C. Knauerhase, D. Barkai, S. Garg, V. Tewari, T. A. Anderson, M. Bowman, "*Toward Internet Distributed Computing*", pp 38-46, vol. 36 , Issue 5, May 2003.

[4]. R. Buyya , D. Abramson , J. Giddy, " *A Case for Economy Grid Architecture for Service Oriented Grid Computing"*, Proceedings of the 10th Heterogeneous Computing Workshop HCW 2001 (Workshop 1), vol. 2, p.20083.1, April 23-27, 2001.

[5]. C. S. Yeo, M. D. Assunçao, J. Yu, A. Sulistio, S. Venugopal, M. Placek, and R. Buyya, "*Utility Computing and Global Grids"*, Grid Computing and Distributed Systems (GRIDS) Laboratory Department of Computer Science and Software Engineering The University of Melbourne, VIC 3010, Australia.

[6]. I. Foster, *"What is the grid? A three-point checklist*", Available, http://www-fp.mcs.anl.gov/~foster/Articles/WhatIsTheGrid.pdf, [Accessed 8th May 2009].

[7]. J. W. Ross G. Westerman, *“Preparing for utility computing: The role of IT architecture and relationship management”*, IBM systems journal, Vol 43, no 1, 2004.

[8]. G. Perry, “How Cloud & Utility Computing Are Different”, http://gigaom.com/2008/02/28/how-cloud-utility-computing-are-different/, [Accessed 12-05-2009].

[9]. L. Dignan, *“Behind the Myths of Cloud Computing”*, http://seekingalpha.com/article/71589-behind-the-myths-of-cloud-computing, [Accessed April 16, 2009].

[10]. A. Weiss, *“Computing in the Clouds”*, Vol. 11, No. 4 ACM New York, NY, USA (2007).

[11]. G. Boss, P. Malladi, D. Quan, L. Legregni, H. Hall, *“Cloud Computing”*, IBM Corporation 2007.

[12]. James Governor, “15 Ways to Tell Its Not Cloud Computing”, http://redmonk.com/jgovernor/2008/03/13/15-ways-to- tell-its-not-cloud-computing/ [Accessed 09-05-2009].

[13]. 3tera, *“Cloudware - Cloud Computing without compromise”*, Available http://3tera.com/Cloud- computing/, [Accessed 19-05-2009].

[14]. T. Eilam et al, *“Using a utility computing framework to develop utility systems”*, IBM SYSTEMS JOURNAL, VOL 43, NO 1, 2004.

[15]. P. Padala, X. Zhu, M. Uysal, Z. Wang, S. Singhal, A. Merchant, K. Salem, *“Adaptive Control of Virtualized Resources in Utility Computing Environments”*, ACM SIGOPS Operating Systems Review, Vol41, No 3. June 2007.

`

[16]. V. Srinivasan, *“The Embedded Quality Framework - A Strategic Approach to Managing Quality of Web Software Products”*, Brassring Inc., Waltham, MA 02453, U.S.A.

[17]. S. Graupner, J. Pruyne, and S. Singhal, *“Making the utility data center a power station for the enterprise grid”*. Technical Report HPL-2003-53, Hewlett Packard Laboratories, March 2003.

[18]. Wohlin. C, Runeson P, H. M, Ohlsson. M. C, Regnell. B, Wesslen. “A Experimentation in Software Engineering: An Introduction”, Kluwer Academic Publishers.

[19]. Web center for social research methods, “*External Validity”*, http://www.socialresearchmethods.net/kb/intval.php, [Access on: 10th May, 2009].

[20]. Web center for social research methods, “External Validity”, http://www.socialresearchmethods.net/kb/external.php, *[Access on: 10th May, 2009].*

[21]. Web center for social research methods, “*Descriptive Statistics”, http://www.socialresearchmethods.net/kb/statdesc.php, [Access on: 10th May, 2009].*

[22]. Dictionary.com: http://dictionary.reference.com, [Accessed 10th May 2009]

[23]. Bondi, Andr´e B., “Characteristics of scalability and their impact on performance”, In: Proceedings of the 2nd international workshop on Software and performance. Ottawa, Canada :

[24]. ACM Press New York, NY, USA, 2000, 195 – 203.