Digital Ecosystems and Business Intelligence Institute
Curtin Business School

SLA-Based Trust Model for Secure Cloud Computing

Mohammed Alhamad

This thesis is presented for the Degree of
Doctor of Philosophy

Curtin University

November 2011

DECLARATION

To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgment has been made.

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

Signature: …….

Date: ……...

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THESIS SUMMARY

Cloud computing has changed the strategy used for providing distributed services to many business and government agents. Cloud computing delivers scalable and on-demand services to most users in different domains. However, this new technology has also created many challenges for service providers and customers, especially for those users who already own complicated legacy systems. This thesis discusses the challenges of, and proposes solutions to, the issues of dynamic pricing, management of service level agreements (SLA), performance measurement methods and trust management for cloud computing.

In cloud computing, a dynamic pricing scheme is very important to allow cloud providers to estimate the price of cloud services. Moreover, the dynamic pricing scheme can be used by cloud providers to optimize the total cost of cloud data centres and correlate the price of the service with the revenue model of service. In the context of cloud computing, dynamic pricing methods from the perspective of cloud providers and cloud customers are missing from the existing literature. A dynamic pricing scheme for cloud computing must take into account all the requirements of building and operating cloud data centres. Furthermore, a cloud pricing scheme must consider issues of service level agreements with cloud customers.

I propose a dynamic pricing methodology which provides adequate estimating methods for decision makers who want to calculate the benefits and assess the risks of using cloud technology. I analyse the results and evaluate the solutions produced by the proposed scheme. I conclude that my proposed scheme of dynamic pricing can be used to increase the total revenue of cloud service providers and help cloud customers to select cloud service providers with a good quality level of service.

Regarding the concept of SLA, I provide an SLA definition in the context of cloud computing to achieve the aim of presenting a clearly structured SLA for cloud users and improving the means of establishing a trustworthy relationship between service provider and customer.
In order to provide a reliable methodology for measuring the performance of cloud platforms, I develop performance metrics to measure and compare the scalability of the virtualization resources of cloud data centres. First, I discuss the need for a reliable method of comparing the performance of various cloud services currently being offered. Then, I develop a different type of metrics and propose a suitable methodology to measure the scalability using these metrics. I focus on virtualization resources such as CPU, storage disk, and network infrastructure.

To solve the problem of evaluating the trustworthiness of cloud services, this thesis develops a model for each of the dimensions for Infrastructure as a Service (IaaS) using fuzzy-set theory. I use the Takagi-Sugeno fuzzy-inference approach to develop an overall measure of trust value for the cloud providers. It is not easy to evaluate the cloud metrics for all types of cloud services. So, in this thesis, I use Infrastructure as a Service (IaaS) as a main example when I collect the data and apply the fuzzy model to evaluate trust in terms of cloud computing. Tests and results are presented to evaluate the effectiveness and robustness of the proposed model.

ACKNOWLEDGEMENTS

I would like to acknowledge the Grace of God for giving me this opportunity to complete my Doctoral Dissertation under the supervision of Professor Tharam Dillon, Professor Elizabeth Chang, and Dr. Farookh Khadeer Hussain. Additionally, I wish to acknowledge the efforts of my family without whose support and sacrifices I would never have been the person that I am today. To my parents, Mr. Ali Alhamad and Mrs. Makkya Alhasan, who have always been there to support me and have sacrificed a lot for me, I can never begin to thank you enough for all that you have done for me. To my wife, Huda Alhaji, my angel girl, Fatema and my angel boys, Ali and Abdullah, I can never thank you enough for your unwavering support and sacrifices made for me. To all my family, I hope that this thesis is a sign of the good things to come in life. Then, I would like to thank my supervisors Professor Tharam Dillon, Elizabeth Chang, Dr. Farookh Khadeer Hussain and thesis committee chair person, Dr Omar Khadeer Hussain for their consistent support, never-ending compassion, encouragement and superb guidance. This thesis is as much a result of their efforts as it is mine. Finally, I dedicate this thesis to Professor Tharam Dillon, who has been a key person throughout the work done in this thesis, my parents, my wife, my daughter, my sons and my family.

LIST OF PUBLICATIONS

Journal Publications

1- Mohammed Alhamad, Tharam Dillon, and Elizabeth Chang, *"A trust-evaluation metric for cloud applications"*, International Journal of Machine Learning and Computing vol. 1, no. 4, pp.416-421, 2011.

2- Mohammed Alhamad, Tharam Dillon, Farookh Khadeer Hussain, *"Performance measurement for cloud services",* International Journal of High Performance Computing and Networking (IJHPCN), under review

3- Mohammed Alhamad, Tharam Dillon, Farookh Khadeer Hussain, "*A dynamic pricing scheme for cloud services*", International Journal of Web and Grid Services (IJWGS), under review

4- Mohammed Alhamad, Tharam Dillon, Farookh Khadeer Hussain, *"Fuzzy-based trust model for cloud computing*", International Journal of Applied Soft Computing, under review

Conferences Publications

1- M. Alhamad, T. Dillon, and E. Chang, *"Conceptual SLA framework for cloud computing",* In Digital Ecosystems and Technologies (DEST), 4th IEEE International Conference on, pp. 606--610, 2010.

2- M. Alhamad, T. Dillon, and E. Chang, *"SLA-based trust model for cloud computing"* Network-Based Information Systems (NBiS), 13th International Conference, pp. 321--324, 2010.

3- A. Mohammed, D. Tharam, W. Chen, and C. Elizabeth, *"Response time for cloud computing providers"*, Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services (iiWAS '10), ACM, pp. 603-606, 2010.

4- Mohammed Alhamad, Tharam Dillon, Chen Wu, Elizabeth Chang, *"A trust-evaluation metric for cloud providers"*, IEEE ICMLC, Singapore, 2011.

5- M. Alhamad, T. Dillon, and E. Chang, *"A survey on SLA and performance measurement in cloud computing",* On the Move to Meaningful Internet Systems, OTM 2011, vol. 7045, Springer Berlin, Heidelberg, pp. 469-477, 2011.

6- Mohammed Alhamad, Tharam Dillon, Elizabeth Chang, *"Service level agreement for distributed services: A Review",* International Symposium on Advances in Cloud and Green Systems (ACGS), Sydney, 2011.

CHAPTER 1 – INTRODUCTION

1.1 INTRODUCTION

Cloud computing offers the benefit of cost-cutting to enterprises by online-allocation of storage and computing resources, because cost depends on whenever and for how long the resources are required. Although cost-effective, this latest technology affects customary security and trust mechanisms employed by these enterprises [1].

Trust can be broadly defined as a mental state in which a person or organization accepts the susceptibility of any process on the basis of positive expectations of the actions of another person or organization [2].

Customers are unable to use technical means to protect their data from secondary usage or illicit access because of their lack of control over cloud resources; therefore, relying on trust-based methods and mechanisms for data security becomes a major issue. Other than trust, there are few other sources that offer reimbursement in case there is a breach of mutual agreement. These may include court action or insurance protection.

Of the different ways of establishing trust, the most important is security. Another component of online trust is reputation. Trust is also related to brand name and image.

The use of cloud computing requires a balance of costs, benefits, security and privacy. Trust is a key to the acceptance of Software as a Service (SaaS). Unless the customer and organizations trust that cloud providers will protect the security of sensitive information, customers may refrain from using cloud services. Secure handling of data, accountability, and privacy safeguards promote trust among users and service providers and also encourage the acceptance of cloud computing services.

In conventional models of Internet security, a security boundary is deployed to build a trust perimeter to control the use of computing resources. In such a model, the customer or organization can control the storage and processing of the data depending on the organizational policies. However, this is not possible with cloud computing because the security boundary is compromised since the data is processed on machines that are owned

and controlled by someone else. The contractors or sub-contractors may also process the classified data independently of the trusted vendors, thereby increasing the risk of illegal use, resale or outflow of sensitive data.

Some of the major issues arising from the use of cloud computing include lack of user control, because as soon as Software as a Service is used, the responsibility for data storage is transferred from the user to the corresponding service provider. Hence, the visibility and control of data is very limited. Another severe risk is that the service provider may obtain profits from unauthorized use of data and since multiple parties are involved, it is very difficult to control the flow of data. There is also a possibility of violation of local laws when transporting data stored in the cloud across geographical country borders. This also makes it difficult to identify the party which is responsible for ensuring that legal requirements are met for personal information and data handling. It is also very complicated to ascertain the trustworthiness of cloud sub-contractors who are involved in processing, particularly in a globalized cloud infrastructure and dynamic environment.

Cloud computing increases the risk that third parties will access private and sensitive data for financial gain. It also offers minimal control over the data lifecycle as the service provider might not delete the data once it has been processed, and holds on to it for unauthorized resale at a later point in time. In order to create a flexible infrastructure, cloud providers might create several backups that produce an increased security risk because these unauthorized backups could lead to severe intimidation from external or domestic attackers. Today's cloud computing also lacks widely-accepted standards such as security, privacy and integrity standards; because of the diverse environments, it becomes difficult for the user to communicate with the vendors. Moreover, exporting data in different formats and setting up security boundaries also becomes cumbersome because of non-standard practices in such diverse environments. Stipulation of a complete audit is also not possible in a cloud computing environment.

Thus, cloud computing has created serious issues and its use is risky unless some important measures are taken to avoid or prevent these risks.

The methodologies proposed in this thesis are intended to reduce such risks in the cloud community. I use the concepts of economics, service level agreements (SLAs), evaluation of performance, and trust management to make several contributions to the cloud computing

domain. These contributions are discussed and evaluated in order to assist cloud providers and cloud customers to use the cloud community to their advantage. More details about my proposed solution are presented in Chapter 4.

In this chapter, I introduce cloud computing and the four main services of cloud providers namely Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Database as a Service (DaaS), and Software as a Service (SaaS). In Section 1.3, I discuss the importance of dynamic pricing schemes for cloud services. Section 1.4 presents the concept of service level agreement in the domain of electronic services. In Section 1.5, the most challenging issues related to trust management of cloud computing will be discussed in order to show the motivation for the main objective of this thesis. Section 1.6 provides the motivation for this research. Section 1.7 presents the objectives of the research. I discuss the scope of this research in Section 1.8 and I show the main research stages which are included in the research timeline. Section 1.9 discusses the significance of developing accurate solutions based on the trust models in order to improve the security and performance of cloud services provided to end users. I present the plan of the thesis and briefly overview the concerns of each chapter in Section 1.10. Finally, I conclude this chapter.

1.2 CLOUD COMPUTING

There have been various definitions proposed in the literature of cloud computing [3-5]. In this research, I adopted and considered the definition provided by the U.S. NIST (National Institute of Standards and Technology): "Cloud computing is a model for enabling convenient, on demand network access to a share pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management afford or service provider interaction" [3]. In other words, cloud computing is a framework in which, by using a virtualized infrastructure, resources are delivered as a service to customers via a public network which is the Internet [6-8].

The cloud customers can range from big organizations, small business, developers or individual users. In this research, such customers will be known as 'users'. One of the advantages of having such a framework is that users do not need to buy costly physical infrastructure or software, but they can use other users' resources over a virtual environment

at far less cost, thereby reducing their own operational and maintenance costs. For example, Salesforce.com developed a customer relationship management solution (CRM) and delivered this as a cloud service, not as a package of software. Salesforce.com customers can access these types of services by using a basic machine with an Internet browser [9] for a fee which is far less than the cost of the package. Figure 1.1 shows the cloud computing stack layers.

There are four main delivery models of cloud services over such a paradigm. They are:

1) Infrastructure as a Service (IaaS): In such architectures, users can use the visualization resources as a fundamental infrastructure for their applications. These resources may be a CPU, network, or storage. Cloud users can manage the resources and assign rules for end users [10].

2) Database as a Service (DaaS): These architectures allow users to rent a specific size of storage for a specific period of time. Users are not required to manage the integration or the scaling of the infrastructure. Database providers take the responsibility for integration, privacy, and security of users' data [11].

3) Platform as a Service (PaaS): Here, users use all facilities on the cloud to develop and deliver their web application and services to the end users. PaaS services may include development, integration, testing or the storage resources required to complete the life cycle of services [12] and other web application. An example of a PaaS is the GoogleApps Engine.

4) Software as a Service (SaaS): With these architectures, users connect with the service providers to use the application, but they do not control the infrastructure, operating system or network infrastructure [12, 13]. An example of an SaaS is Google Docs.

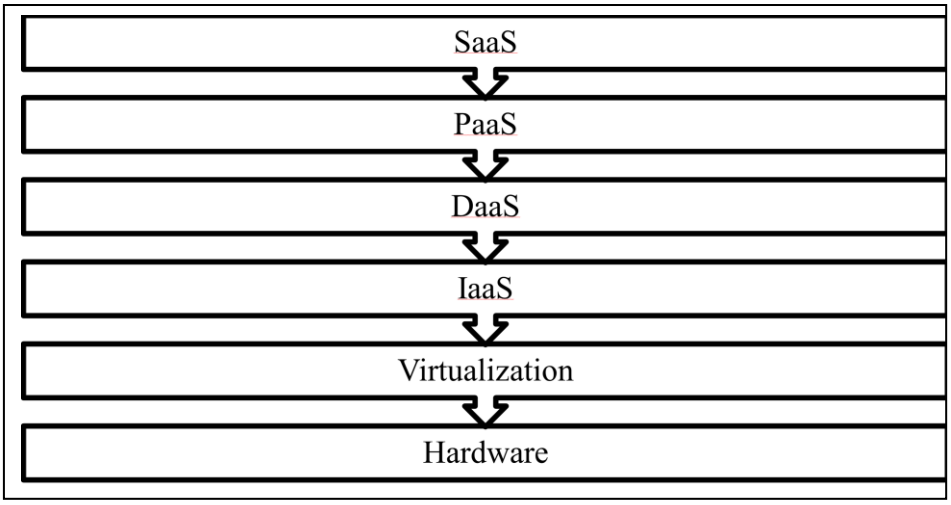


Figure 1.1: Cloud computing stack layers

No matter what type of delivery model is being used, there are five essential factors or characteristics that have to be satisfied in order to achieve smooth computing in a cloud computing environment. They are:

a) *On-demand self-service:* On demand self-service refers to the availability of the required resources (such as CPU power, network etc) as and when the user needs them. Furthermore, this should not require any human intervention [14].

b) *Broad network access:* As the interacting medium between the different users is the Internet, there should be a broad network access available that allows for seamless interaction of different applications across different heterogeneous platforms [15].

c) *Resource Pooling:* A cloud provider should support multi-tenancy of its resources for maximising the efficiency of its infrastructure. For example, it should be able to dynamically assign the required resources to the consumer according to its demand [16].

d) *Rapid Elasticity:* It should be flexible according to the computing resources required by the customers. For example, there is no up-front commitment and the customers should be able to release the resources once their work is over [17].

e) *Measurement of Service Consumption:* There should be a framework that measures the usage of each user according to the resources that are being used by it [18].

1.3 PRICING SCHEMES

The pricing of electronic services poses a significant challenge to service providers. The delivery of services via the Internet becomes an attractive model for businesses. An accurate prediction scheme of dynamic-based pricing helps to maximize the revenue of the service provider and provides competitive prices for service customers. In cloud computing, the difference in the consumption of resources between peak and off-peak periods of demand is very significant. Cloud data centres are administered using a huge investment budget; the optimization of service resources allocation for customers is crucial to the service provider's business success and longevity. In order to ensure the optimization goal for resource allocation, a dynamic pricing scheme has to be used with the correlation method when creating a costing model for cloud data centres. The dynamic pricing scheme is used widely

in different service domains. Examples of these domains are network services, peer-to-peer, grid, and electronic services. In the literature, many of the proposed approaches to dynamic pricing are unable to show a relationship between pricing service and costing parameters of service resources. Using a pricing model in this case increases the business' level of risk in terms of its continued viability.

In this thesis, I discuss the problem of dynamic demand in terms of the cloud market and propose a methodology for dynamic pricing in the domain of cloud computing. More details are presented in Chapter 5 of this thesis.

1.4 SERVICE LEVEL AGREEMENTS

A Service Level Agreement (SLA) [19-21] is a contract that describes the agreed service, service level parameters, guarantees, and actions and consequences for all cases of violations [22]. The SLA is very important as a contract between consumer and provider. The main idea of SLAs is to clarify and formalize the agreements about service terms such as performance, availability and billing. It is important that the SLA include the obligations and the actions that will be taken in the event of any violation, with clearly shared semantics between each party involved in the online contract. The SLA is a legal format documenting the way that services will be delivered as well as providing a framework for service charges. Service providers use this foundation to optimize their use of infrastructure to meet signed terms of services. Service consumers use the SLA to ensure the level of quality of service they need and to maintain acceptable business models for long-term provision of services. In the current literature, there is no mechanism for the formulation of SLAs in the cloud environment. Furthermore, there is no framework to customize the SLAs based on the type of cloud service (IaaS, PaaS, DaaS, SaaS) being used. In this thesis, I propose a methodology to use SLAs with suitable customized metrics that can be deployed in cloud services which are IaaS, PaaS, DaaS, and SaaS. More details are presented in Chapter 6.

1.5 TRUST AND REPUTATION

The technology of the Internet, service-oriented architecture, and new paradigm of cloud computing provide great advantages for online users because of their ease of connectivity. This has presented new features of technology like dynamic, multi tenants, and schemes for dynamic pricing. However, these technologies introduce many challenges for online users and e-business agents. One of the significant challenges is the trust management issue[23]. Recently, the number of users of online services has increased significantly. These users perform multi-purpose activities on the Internet and present high risks for both other users and service providers. Users of online services perform their transactions in anonymous, pseudo, and non-anonymous forms of environments. In this type of community, users may perform many malicious or unethical activities to harm the competitors or the service provider. Also, service providers may provide many services with violations of the agreed terms of services. Trust concepts are very important for all participants in online communities to regulate the above mentioned non-complying behaviours. Using trust in an appropriate way can reduce the risks related to transactions taking place in an anonymous environment and ensure a good level of soft security for online users and service providers. In cloud computing, the use of trust technology is highly advantageous. By combining trust solutions with the SLA concept, users of cloud services can guarantee quality of services without having to invest huge sums to build complicated solutions along the lines of traditional security.

In Chapter 3, I formally define this problem and discuss in detail the issues and solutions that can be implemented with trust and reputation concepts in order to enhance the level of security in cloud computing.

1.6 MOTIVATION

Trust and measurement of the quality of cloud services becomes an important issue as the infrastructure and management of distributed services switches from traditional computing centres to public data centres. Customers of cloud services require a high level of security, performance, and privacy to ensure the continuity of their business. AMD [24] conducted an extensive, comprehensive survey in which they interviewed selected businesses who are interested in using cloud services. All expressed considerable concern about issues of security

and privacy. Figure 3 compares risks of cloud business services and indicates the importance of security and privacy issues as the first consideration of interviewee organizations.

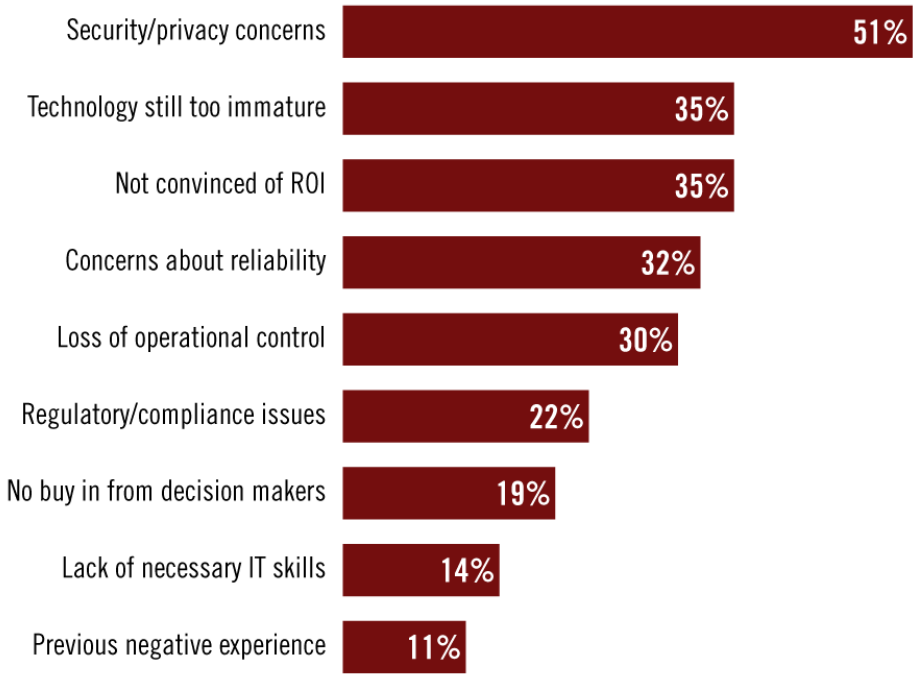


Figure 1.2: Indication of importance of security and trust issues in public and private sectors (Source: AMD [24])

The number of companies and government agencies who use cloud services are increasing. Billions of dollars have been spent over the last few years on moving the existing data storage from traditional data pools to outside the organizations' borders. Most of the data are sensitive and include huge amounts of personal data of customers and residents. Cloud infrastructures which are implemented to provide a sufficient level of security and privacy should deploy reliable and high performance approaches, which should comprise much more than the current security solutions. In this thesis, I propose an SLA-based trust framework to enhance security and minimize the risks of using cloud services.

Furthermore, large organizations and e-government agents are becoming increasingly interested in using the new technology of cloud services. It is important to measure the performance of infrastructure, networking, and applications of cloud platforms in order to guarantee some level of reliability and security of cloud services. Since the announcement of

public cloud services such Amazon EC2, Google Apps, and Microsoft Azure, there has emerged the need for a framework that measures and assesses the performance of cloud platforms. Such a framework should be implemented to fulfil the requirements of cloud users. These requirements include dynamic monitoring of performance of virtualized machines, internal networking, wide networking, storage infrastructure, and costing metrics. This will enable cloud users to use the results of measurements to estimate the quality of cloud services and their cost without deploying any application on a selected cloud platform. In this thesis, I define a methodology to help cloud users compare cloud platforms and use the results of my method to select the service provider who is most likely to meet the service requirements.

1.7 OBJECTIVES OF THE THESIS

The main objective of this thesis is to develop an SLA-based trust model for cloud providers. This model can be used to enhance the security and reliability of cloud services. The proposed model will be developed as a trust-based framework to be used by cloud users. The main objective of this research can be broken down into 6 different sub-objectives. They are:

1) To develop a dynamic pricing model for cloud services. This model aims to solve the problem of how to correlate the proposed price of cloud service with the costing level of cloud resources.

2) To investigate and analyse the main requirements to establish an effective model for SLA in cloud computing.

3) To compare the most important performance metrics of the current cloud providers such as Amazon EC2, Google Docs, and Microsoft Azure by using different types of benchmarks such as the scalability of CPU, storage disk, and network infrastructure.

4) To develop a trust model for the cloud computing community that will assist users to choose the most secure and reliable services from a large number of cloud services.

6) Simulation and validation of the solutions developed in (1) - (4) above.

1.8 SCOPE OF THE THESIS

This thesis presents and validates a basic methodology to help both cloud providers and their users to establish a trusted relationship and perform the business tasks in a more secure environment. The main focus of this thesis is to develop methodologies for dynamic pricing, SLAs, performance measurement and trust management for cloud computing. I do not focus on how to manage the data, develop cloud communication protocols, or provide security applications for the cloud infrastructure, even though I consider my proposed solution as a sub-solution of the large integrated security solutions for the cloud environment. In the SLA designing phase of this thesis, the main work is on the design of a clear format for cloud service agreements and I demonstrate how to select the most important parameters which must be included in such agreements. To do that, I review the existing SLAs of different cloud services providers including Amazon EC2, Microsoft Azure, and Salesforce. Also, my scope includes the development of a methodology to collect the cloud services' measured values in order to use them as one of the main inputs for a fuzzy-based trust system to compute the trustworthiness and credibility of cloud providers. In the validation and evaluation phase of this thesis, I focus on an example of cloud services platforms to compare my outcomes with one of the existing methods of trust calculation in distributed architecture.

1.9 SIGNIFICANCE OF THE THESIS

The significance of this research can be divided into two paradigms which are social and scientific.

The social significance of this research includes:

1. Reducing the risk when cloud users move to cloud platforms.

2. Developing a trust model to enhance the privacy solutions provided by cloud data centres.

3. The proposed model can be improved to extend its application to online learning, biomedical, and other social communities that have already developed cloud computing.

4. The proposed model can be used as a basic tool for measuring the performance metrics of cloud services such as response time, throughput, and visualization resources.

The scientific significance of this research includes:

1. Investigating the feasibility of using SLAs to constrain cloud providers to deliver high quality of services. The proposed definitions of the most important SLAs for the different types of cloud services can be technically monitored. Based on the results of the monitoring process, the values of trust for each participant provider can be scaled and ranked in order to improve the quality of services being offered

2. The proposed model can be used to rank the cloud providers based on the trust values. The trust model that will be designed considers the trust level of the cloud services and also the general reputation of providers themselves. This research will use different sources of trust data such as the number of violations, user's feedback, and the quality of services to develop a reliable weighting formula in order to assign a fear trust value to cloud providers

3. Performance evaluation of the most common cloud platforms. In this research, the measurement approach will be developed to compare different cloud platforms such as Amazon EC2 platform, Google Docs, and other popular cloud providers

4. This research will develop a clear pricing model for cloud computing, and cloud providers can use this model to define SLA parameters which improve their profit and provide reliable services for the customers who use the cloud services. As mentioned in Section 2, customers include government departments, social users, developers, organizations, small business or personal users.

Research work into (1) – (4) above presents general work in the respective domain of cloud computing.

1.10 PLAN OF THE THESIS

Chapter 2: This chapter presents the related literature and reviews the existing solutions to the trust-related problems in different domains such as distributed computing and service-oriented architecture. Also, the issues associated with designing service level agreements, performance measurements, and pricing models for electronic services will be reviewed to illustrate the main problems which this research is intended to solve.

Chapter 3: Chapter 3 defines the thesis problem and provides succinct definitions of terms which I use in the research problem. Also, in this chapter I discuss a number of research methodologies that are used to solve various problems in the cloud computing domain, and briefly explain the suitable method.

Chapter 4: Chapter 4 describes the proposed solution to the research problem which is presented in Chapter 3. Also, this chapter links the research problem to each solution which is discussed in detail from Chapter 5 to Chapter 8.

Chapter 5: The dynamic pricing scheme for cloud computing services is presented in this chapter. In order to design a more durable SLA for cloud users, users of cloud services need to know how they will be charged. Cloud services consumers will use the pricing scheme proposed in this chapter to compare the quality of cloud services with the associated costing parameters of the cloud provider who provides the service. The pricing scheme focuses on how the service provider can estimate the price of cloud services prior to signing the SLA between users.

Chapter 6: This chapter presents the main criteria which should be considered when designing the SLA in cloud computing. A well-defined structure for cloud service SLAs is presented in this chapter for each of the four different types of cloud services (IaaS, PaaS, DaaS, and SaaS).

Chapter 7: In this chapter, I present a methodology to evaluate cloud services in order to help cloud users select the most reliable resources. I conduct real-world experiments on Amazon EC2 platform to present a new solution for defining the reliable criteria for the selection process of cloud providers.

Chapter 8: In this chapter, I provide a basic methodology that will assist both cloud providers and their users to establish a trusted relationship and perform the business tasks in a more secure environment. The main focus of this chapter is to develop a model which uses the fuzzy logic technique with performance measured parameters to carry out the trust values about cloud service providers. The SLAs for cloud users, cloud services performance, and the fuzzy logic approach are used to develop the proposed trust model for the cloud services environment.

Chapter 9: This chapter concludes my research work and provides directions for future work.

1.11 CONCLUSION

In this chapter, I provide an overview of the cloud computing services and the main differences are pointed out in order to clarify the model of cloud services offered by current providers. Then, the importance of a dynamic pricing model of cloud services is discussed. I present a brief description of the service level agreement concept. Then, I discuss the trust issues pertinent to cloud computing.

Also, the motivation for this research is briefly presented, followed by thesis contributions and objectives. Then, the limitations and scope of the thesis are discussed to give an idea about the focuses in the research domain of cloud services. Finally, the structure of the thesis is described and I discuss the main points of the research methodology for each chapter of the thesis.

REFERENCES

[1] Lori M. Kaufman, *"Data security in the world of cloud computing"*, IEEE Security and Privacy, v.7 n.4, pp. 61-64, 2009.

[2] Rousseau D.M, Sitkin S.B, Burt R.S, and Camerer C, *"Not so different after all: A cross-discipline view of trust"*, Academy of Management Review, 23, 3, pp. 393-404, 1998.

[3] P. Mell and T. Grance, *"Draft nist working definition of cloud computing"* Referenced on June. 3rd, 2009 Online at http://csrc.nist.gov/ groups/SNS/cloud computing/index.html, 2009.

[4] J. Napper and P. Bientinesi, *"Can cloud computing reach the top500?"*, UCHPC-MAW, Proceedings of the combined workshops on UnConventional High Performance Computing Workshop Plus Memory Access Workshop. New York, USA: ACM, pp. 17-20, 2009.

[5] Y. Chen, V. Paxson, and R. Katz, *"What's new about cloud computing security?"* Technical Report UCB/EECS-2010-5, UC Berkeley Department of EECS, 2010, http://www.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS-2010-5.pdf, Accessed on Jan 2010.

[6] R. Buyya, *"Market-Oriented Cloud Computing: Vision, Hype, and Reality of Delivering Computing as the 5th Utility"*, IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID'09), Shanghai, China, May, pp. 1, 2009.

[7] A. Marinos and G. Briscoe, *"Community Cloud Computing",* Lecture Notes in Computer Science, Vol. 5931/2009, pp. 472-484, 2009.

[8] P.T. Jaeger, T. Lin, and J.M. Grimes, *"Cloud computing and information policy: computing in a policy cloud"*, Journal of Information Technology & Politics, vol 5(3), pp. 269-283, 2008.

[9] M. Nelson, "Building an Open Cloud," *Science*, vol. 324, pp. 1656, 2009.

[10] D. Hilley*, "Cloud Computing: A Taxonomy of Platform and Infrastructure-level Offerings"*, Technical Report 13, Center for Experimental Research in Computer Systems - Georgia Institute of Technology, 2009.

[11] H. Cai, K. Zhang, M. Wang, J. Li, L. Sun, and X. Mao, "*Customer Centric Cloud Service Model and a Case Study on Commerce as a Service"*, IEEE International Conference on Cloud Computing, pp. 57-64, 2009

[12] D. Cerbelaud, S. Garg, and J. Huylebroeck, *"Opening the clouds: qualitative overview of the state-of-the-art open source VM-based cloud management platforms"*, 10th ACM/IFIP/USENIX International Conference on Middleware, pp. 22, 2009.

[13] J. Müller, J. Krüger, S. Enderlein, M. Helmich, and A. Zeier, "*Customizing enterprise software as a service applications: Back-end extension in a multi-tenancy environment"*, 11th International Conference on Enterprise Information Systems (ICEIS), Lecture Notes in Business Information Processing, vol. 24. Springer, pp. 66-77, 2009.

[14] B. Sotomayor, R. Montero, I. Llorente, and I. Foster, *"Virtual infrastructure management in private and hybrid clouds"*, *IEEE Internet Computing*, vol. 13, pp. 14-22, 2009.

[15] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov, *"The eucalyptus open-source cloud-computing system"*, UCSD Tech. Rep, pp. 124-131, 2009, http://eucalyptus.cs.ucsb.edu/, Accessed on December 2009.

[16] M. Zeller, R. Grossman, C. Lingenfelder, M. Berthold, E. Marcade, R. Pechter, M. Hoskins, W. Thompson, and R. Holada, *"Open standards and cloud computing: Kdd-2009 panel report"* pp. 11-18, 2009

[17] T. Dillon, C. Wu, and E. Chang, *"Cloud Computing: Issues and Challenges"*, IEEE Int'l. Conf. Advanced Info. Networking and Apps, p. 27–33, 2010.

[18] J. Nunamaker Jr, M. Chen, and T. Purdin, *"Systems development in information systems research,"* Journal of Management Information Systems, pp. 89-106, 1990.

[19] M. Boniface, S. C. Phillips, A. Sanchez-Macian, and M. Surridge, *"Dynamic Service Provisioning Using GRIA SLAs"*, Service-Oriented Computing-ICSOC, International Workshops, Vol. 4907, Springer, pp. 56–67, 2007

[20] G. Di Modica, O. Tomarchio, and L. Vita, *"Dynamic SLAs management in service oriented environments"*, The Journal of Systems & Software, vol. 82, pp. 759-771, 2009.

[21] A. Keller and H. Ludwig, *"The WSLA framework: Specifying and monitoring service level agreements for "*, Journal of Network and Systems Management, vol. 11, pp. 57-81, 2003.

[22] A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, J. Pruyne, J. Rofrano, S. Tuecke, and M. Xu, *" agreement specification (WS-Agreement)"*, Technical Report, Grid Resource allocation Agreement Protocol (GRAAP)-WG, 2004.

[23] P. Resnick and R. Zeckhauser, *"Trust among strangers in Internet transactions: Empirical analysis of eBay's reputation system"*, Advances in Applied Microeconomics: A Research Annual, vol. 11, pp. 127-157, 2002.

[24] AMD, *"Adoption, approaches & attitudes"*, the future of cloud computing in the public and privates sectors", Research Report, Red Shift, 2011, http://www.amd.com/us/Documents/Cloud-Adoption-Approaches-and-Attitudes-Research-Report.pdf , Accessed on July 2011.

CHAPTER 2 – LITERATURE REVIEW

2.1 INTRODUCTION

Cloud computing has changed the strategies for providing distributed services to many businesses and government agencies. Cloud computing delivers a scalable and on-demand service to most users in different domains. This new technology poses many challenges for service provider and customer, especially for those users who already own complicated legacy systems. This thesis examines challenges related to the concepts of trust, SLA management, pricing of cloud services and performance measurement of cloud. I start with a survey of cloud computing architecture. Then, I discuss existing frameworks of service level agreements in different domains such as Web Services and grid computing. The last part of the literature review discusses the advantages and limitations of performance measurement models in SOA, distributed systems, grid computing, and cloud services. Finally, I summarize and conclude my work on the literature review.

2.2 CLOUD COMPUTING AND OTHER DISTRIBUTED SYSTEMS

There has been active research into cloud computing since late 2007. Before cloud, there was grid technology. Now, the hot topic of research is cloud and several more proposed frameworks and models of various new technology solutions have started to be applied to the cloud architecture. In this section, I survey the literature to find the most appropriate definition for the term ‘cloud computing’. Also, I review the different architecture frameworks and the common challenges that might present major problems for providers and customers who are interested in understanding this type of distributed computing.

The Google trends report shows that cloud computing surpassed grid computing in late 2007.

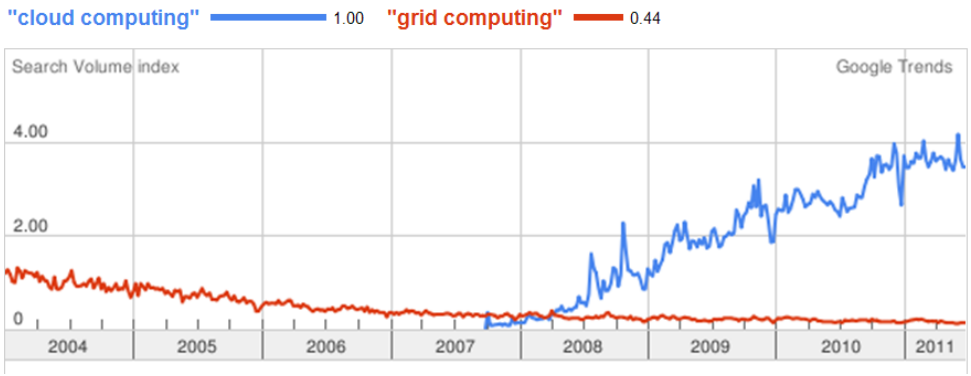


Figure 2.1: Cloud computing trend, source, Google search engine

2.2.1 Definition

Experts and developers who investigate issues and standards related to cloud computing do not have the same background in technology. In the research projects, professionals from grid technology, SOA, business, and other domains of technology and management domains proposed several definitions of cloud computing. However, these definitions are not comprehensive or standard enough to cover most of the technology and other aspects of cloud computing architecture.

In the context of networking and communication, the term “cloud” is a metaphor for the common internet concept [1]. The cloud symbol is also used to present the notion of network connection and the way that the cloud technology is provided by the Internet infrastructure. “Computing” in the context of cloud domain refers to the technology and applications that are implemented in the cloud data centres[2].

Vaquero et al. [3] highlight the lack of a common definition of cloud computing. They state that developers and business decision makers confuse an understanding of the technology with the features of cloud data centres. So, large budgets are often allocated to implement private or even public cloud data centres. However, these data centres face several problems when users or public customers want to connect the interfaces of their legacy systems with the new technology of cloud architecture. Vaquero et al. [3] link the challenge of maximizing the revenue of building cloud technology to professionals who are involved in distributed services. Because they come from a traditional computing domain, they have been confused about the other concepts of distributed services such as Grid, and Web Services. The definition used by Vaquero et al. [3] is as follows:

"Clouds are a large pool of easily usable and accessible virtualized resources (such as hardware, development platforms and/or services). These resources can be dynamically re-configured to adjust to a variable load (scale), allowing also for an optimum resource utilization. This pool of resources is typically exploited by a pay-per-use model in which guarantees are offered by the Infrastructure Provider by means of customized SLAs".

Although, this definition presents the main features of cloud computing, it does not include important components of cloud architecture such as the method of establishing and managing network, applications, and supporting services.

Wang [4] defines cloud computing as: *"A computing Cloud is a set of network enabled services, providing scalable, QoS guaranteed, normally personalized, inexpensive computing infrastructures on demand, which could be accessed in a simple and pervasive way".*

Wang's definition of cloud focuses on the technical aspects of services and does not include the business and functional characteristics. On other hand, Gruman and Knorr [5] explain the main technical concepts of a cloud services model and define cloud computing from the developer's point of view. The authors show how the cloud computing architecture takes advantage of the way that different distributed services are implemented, mainly SOA. Two types of cloud services are included in this definition: SaaS and PaaS. Despite the importance of IaaS as a main component of cloud architecture, they do not adequately discuss this type of cloud delivery model.

In this research, I adopt and use the definition provided by the U.S. NIST (National Institute of Standards and Technology) [6] that defines cloud computing thus: *"Cloud computing is a model for enabling convenient, on demand network access to a share pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management afford or service provider interaction"* [6].

The shortcomings of above proposed definitions of cloud computing are:

1. None of the above definitions includes the technical and business aspects of cloud computing. So, decision makers in large organizations are confused, especially when they want to define the parameters of a costing model for cloud services.
2. Existing cloud definitions do not specify the onus of responsibility in cases where a low level of QoS is delivered
3. Most of the proposed definitions consider a specific type of cloud service, whereas a definition of cloud needs to encompass all classes of cloud services.
4. The proposed definitions do not include or specify the potential cloud users.

Table 2.1 shows the scope of these definitions and lists their main shortcomings:

Table 2.1: Summary of several cloud computing definitions

Reference	The scope of definition	Missing
Vaquero [3]	Defines architecture and service model	Management, supporting, and trust concepts
Wang [4]	Technical concepts	Business and functional characteristics
Gruman [5]	Compares cloud computing with Web Services, and SOA	Definition of IaaS and DaaS
Mell [6]	Technical features, management, and security concepts. This definition is adopted in this thesis to define the cloud computing architecture.	Costing and billing model

2.2.2 Taxonomy of Cloud Computing

Buyya et al. [7] present more than fifteen characteristics which distinguish cloud computing from other distributed systems. Buyya uses scalability, automatic adaption, virtualization, and a dynamic model of billing as the main concepts that constitute the architecture of cloud computing. Moreover, he explains the means of delivering cloud services to different types of users. For instance, users who want to develop small size applications can connect to one of the PaaS such as Microsoft Azure [8],without having to install any of the development tools. Hoefer [9] presents a clear taxonomy framework for the existing category of cloud services. The class of cloud services is described in a tree-structured taxonomy, and the unique characteristics of each model of service are used to identify each node of the proposed tree structure. Hoefer's system of classification provides a clear comparison of cloud services at a high level on the tree structure. However, at the lower end of the structure, the taxonomy of cloud services is not enough to distinguish the various types of services in more detail. The taxonomy presented by Laird in [10] defines the cloud technology from the perspective of service providers. The proposed taxonomy includes the common vendors of cloud services. Laird classifies services according to two categories. The first defines the infrastructure of cloud services, and the second defines the services based on cloud features such as security, billing, and applications which are built on the system. Rimal et al. [11] present a comprehensive framework for the architecture of cloud computing. They describe the taxonomy of cloud services with more focus on the management domain of cloud contents. The concepts of management, business, billing, and support of cloud services are described in depth in order to present the cloud architecture as a new business model. The main advantage of the proposed work by Rimal is that relationships between security features and cloud components are provided as a part of the comparison of service models in cloud computing. The taxonomy proposed by Oliveira et al. [12] classify the concepts of cloud computing according to the dimensions of cloud architecture, business model, technology infrastructure, pricing, privacy, and standards. The proposed taxonomy is provided in a hierarchical tree with parent and child relationships. Oliveira uses SaaS, PaaS, IaaS, and DaaS as sub-taxonomy for the business model. This classification is used in the literature of cloud computing to distinguish the service delivery for end users of cloud services. However, these sub-taxonomy terms may create confusion in understanding the way that various business models for cloud services are built. The taxonomy proposed by Oliveira describes the

concepts of cloud architecture from the perspective of e-science. Therefore, many technical aspects of cloud computing are missing in the proposed taxonomy.

2.3 PRICING SCHEMES IN ELECTRONIC SERVICES

In the beginning, users use small amounts of internet resources. To increase the market, a unified price package is used. In these types of pricing models, for a certain time period, the users have the same access speed and price. This is a simple system where the charges are known to the users and this system is quite common in the Internet services access market [13].

There are, however, some disadvantages as well. The users do not have any idea about how to adapt their usage models, and they overuse the online resources. Then service providers treat all the users in the same way by pricing regardless of their individual usage. This does not increase overall development of the distributed technology, and the performance of the system is degraded. As the technology is developed further (in areas such as online applications and the complex e-commerce environment), this model no longer applies.

The resource-based pricing scheme is proposed in [2]. This model charges the users independently. The charges are calculated on the basis of the amount of transferring and the receiving size. The service provider uses statistical sampling methods to assess the usage. The charges are according to the demands of the supply. The interaction between service customers and service providers is also developed, unlike for the fixed pricing model. It enhances the consuming approach efficiency of service resources and controls the bottleneck of the usage. But as with the development of high-bandwidth required services, the overall users' allocation of resources has increased. The charges for the users have increased as well. In terms of content sharing in some charging methodologies, this has become a problem for online-based service providers to charge according to the consumption of resources.

Authors in [14] suggested a complicated pricing model. In this approach, customers prefer traditional services at a fixed rate, but demands for more resources are charged on the basis of consumption. The analysis of the simulation results shows that the complicated price modelling approach can lead to an improvement in network resource performance while enhancing the service provider revenue [14].

In [3] authors proposed a blocking pricing scheme. With the problems of high resource applications, privacy issues and extra costing problems, network blocking has increased. This has created a negative effect called ‘cost-based allocation’ for the service users. So a new model, ‘congestion scheme’ of pricing, has been developed [15] which groups the price so that it can reflect the service resource usage and the service cost. It encourages the users to define rules for the service demand. In this way, blocking problems are addressed. The price is dynamically determined according to congestion.

But the measurement of cost-based allocation is difficult, as each user’s value of service resources is different. A dynamic pricing approach was proposed to measure the cost-based allocation. Several other methods have also been introduced, and the main goal is to determine a price aware service resource system that shifts the amount of load from a time of high load to a time when service resources are stable. In this way, the blocking of traffic can be controlled to some level of reliability [15].

2.4 SERVICE LEVEL AGREEMENTS IN CLOUD ENVIRONMENTS

A service level agreement is a document that includes a description of the agreed service, service level parameters, guarantees, and actions for all cases of violation. The SLA is very important as a contract between consumer and provider. The main idea of SLAs is to give a clear definition of the formal agreements about service terms such as performance, availability and billing. It is important that the SLA include the obligations and the actions that will be taken in the event of any violation, with clearly shared semantics between each party involved in the online contract.

This section of the literature review discusses works related to SLAs in three domains of distributed services. First, I discuss the proposed SLAs structure for Web Services. Second, the frameworks of SLAs designed for grid computing are reviewed. Third, I discuss the main works that pertain specifically to cloud computing. Finally, I list the main shortcomings of SLAs frameworks which are described in this section.

2.4.1 SLAs for Web Services

Several specifications for defining SLAs have been proposed for Web Services. WSLA language [16] introduces a mechanism to help users of to configure and control their resources in order to meet the service level. Also, the service users can monitor SLA parameters at run-time and report any violation of the service. WSLA was developed to describe services under three categories: (a) Parties: in this category, information about service consumers, service providers, and agents is described. (b) SLA parameters: here, the main parameters which are measurable parameters are presented as two types of metrics. The first is resource metrics, a type of metrics used to describe service provider's resources as row information. The second one is composite metrics. These metrics are used to represent the calculation of the combination of information about a service provider's resources. The final category of the WSAL specification is Service Level Objective (SLO). This section is used to specify the obligations and all actions when service consumers or service providers do not comply with the guarantees of services. WSLA provides an adequate level of online monitoring and contracting but does not clearly specify when and how a level of service can be considered a violation. WSOL [17] is a service level specification designed mainly to specify different objectives of Web Services. Concepts of service management, cost and other objectives of services can be presented in WSOL. However, WSOL is not comprehensive enough to be used with the objectives of the new paradigm of cloud computing.

A WS-Agreement [18] is created by an Open Grid Forum (OGF) in order to produce an official contract between service consumers and service providers. This contract should specify the guarantees, the obligations and penalties in the case of violations. Also, the functional requirements and other specifications of services can be included in the SLA. There are three main sections in the WS-Agreement: name, context, and terms. A unique ID and optional names of services are included in the name section. The information about service consumer and service provider, domain of service, and other specifications of service is presented in the context section. Terms of services and guarantees are described in more detail in the terms section. These types of online agreements were developed for use with general services. For cloud computing, service consumers do not have more specific solutions for SLA which present the main parameters of the visualization environment; at the same time, these solutions should be dynamically integrated with the business rules of cloud consumers.

The primary shortcomings of these approaches is that they do not provide dynamic negotiation mechanisms, and various types of cloud consumers need a different structure of implementation of SLAs to integrate their own business rules with the guarantees that are presented in the targeted SLA.

2.4.2 SLAs for Grid Computing

In the context of grid computing, there are a number of proposed specifications which have been developed specifically to enhance different dimensions related to security and trust for grid services. Sahai et al. [19], propose an SLA-based knowledge domain to represent the measurable metrics for business relationship between all parties involved in the transaction of grid services. Also, the authors proposed a framework to evaluate the management proprieties of grid services in the lifecycle. In this work, business metrics and the management of an evaluation framework are combined to produce an estimated cost model for grid services. The framework proposed in this work lacks a dynamic monitoring technique to help service customers know who is taking responsibility when some level of service level is not met as stated in SLA documents. In my research, I extend this approach in order to build a general costing model based on the technical and business metrics of the cloud domain. Leff et al. [20] provide a study about the main requirements to define and implement SLAs for the grid community. The ontology and a detailed definition of grid computing are provided. Then, a scientific discussion is presented about the requirements that can help developers and decision makers deploy trusted SLAs in a grid community. The author implemented a basic prototype in order to validate the use of SLAs as a reliable technique when the grid service provider and customer need to build a trustworthy relationship. The implementation of the framework in this study does not consider important aspects of security and trust management in grid computing. Keung [21] proposed a SLA-based performance prediction tool to analyse the performance of grid services. Keung uses two sources of information which are the main inputs for the proposed model. The source code information and hardware modelling are used to predict the value of performance metrics for grid services. The model proposed by Keung can be used in other types of distributed computing. But in the cloud environment, this model cannot be integrated with a dynamic price model of cloud services. It needs to be improved by using different metrics for cost parameters to reflect the actual price of cloud services. The system proposed by Padget in [22] considers the response time of applications in the grid systems. The main advantage of the proposed system is that it can predict the CPU time for any node in the grid network before conducting the execution.

Padget tested the adaptation SLA model using real experiment on the grid. The prediction system gives values for response time closely to the values when users execute the same application on the grid. Regarding the delay recorded for the large executed files, the author considers that the delay is due to the external infrastructure such as internet connections. The author also discusses the impact of the time delay caused by external parties to the reputation of service providers in case of using SLA management systems. Although, the author provides an efficient method for calculating the response time for grid resources, this work does not include metrics such as security and management metrics.

2.4.3 SLAs for Cloud Computing

The context of this research focuses on service level agreement management in cloud communities. In the sections above, I present frameworks and models in the literature that are mainly designed for managing SLAs in traditional distributed systems. In this section, SLAs and approaches to agreements and negotiations in the cloud community are presented.

Valdimir [23] describes the quality of services related to cloud services and different approaches applied to map SLAs to the QoS. Services ontology for cloud computing is presented in order to define service capabilities and the cost of service for building general SLAs framework. The proposed framework does not consider all types of cloud services; it is general and was tested on the Amazon EC2 only; other types of cloud providers such as PaaS, DaaS, and SaaS should be considered.

The framework developed by Hsien [24] focuses on Software as a Service model of delivery in cloud computing only. Further details are provided on how the services can be integrated to support the concept of stability of the cloud community, especially for SaaS. It fails to consider other providers of services such as PaaS, DaaS and IaaS.

Shortcomings of the proposed works for SLAs in the context of distributed services

The frameworks and structures that are discussed in previous sections present the following problems:

1. The existing frameworks focus more on the technical attributes than on the security and management aspects of services.

2. The proposed structures of SLAs in the above domains need a clear definition of relationships between level of violations and the cost of services.
3. Most of the above studies do not integrate a framework of trust management of service provider with the collected data from monitoring systems of SLAs.
4. The concepts and definitions of service objectives and service descriptions included in SLAs are not easy to understand, especially by business decision makers.
5. The proposed works for cloud environment focus more on the evaluation of virtualization machines on local servers than on existing cloud service providers.
6. Most of the proposed structures of SLAs are defined by technical experts.

No existing work on defining SLAs in cloud computing takes into account all the four different types of cloud services, namely IaaS, PaaS, DaaS and SaaS.

2.5 PERFORMANCE MEASUREMENT MODELS

Cloud providers have increased to deliver different models of services. These services are provided at different levels of quality. Cloud customers need to have a reliable mechanism to measure the trust level of a given service provider. Trust models can be implemented with various measurement models of services. As a part of this research, I investigate the use of a measurement approach in order to develop a general trust model for the cloud community. In this section, performance measurement models of SOA, distributed, and grid services will be reviewed.

2.5.1 SOA Performance Models

Kounev et al. in [25] identify an analytical approach for modelling performance problems in SOA based application. The authors discuss the different realistic J2EE application for large systems of SOA architecture. The validated approach has been tested for capacity planning of the organizations that use distributed services as outsourcing infrastructure. The advantage of the proposed method is its ability to predict a number of application servers based on the collected information of SLA metrics. Walter et al. [26] implemented a simulation tool for analysing the performance of composite services. They used an online book store as a case study to simulate experimental scenarios. They focused on measuring communication latency and transaction completion time. Real data sets were compared with the simulation results,

and the authors state that the simulation tool presents a result that is close to the real data. This type of simulation can be extended and applied to other distributed services. For cloud computing, more effort is required to make this technique compatible with existing interfaces of cloud providers. Rud et al. in [27] use WS-BPEL composition approach to evaluate the performance of utilization and throughput of SOA-based systems in large organizations. They developed the proposed methodology with a mathematical model in order to improve the processes of service level agreements in the SOA environment. The main objective of Rud's method focuses on the management aspects of services. However, performance issues of response time, data storage, and other metrics of technical infrastructure are not considered in this approach. The optimization of total execution time and minimization of business process costs, Menasce in [28] provides an optimized methodology based on the comparison of performance metrics of SOA-based services. In this study, Menasce develop the proposed method to estimate cost level of all services registered in the SOA directory for medium size organizations. Then, the cost metrics is compared with the actual performance of services. The parameters of the performance metrics can be selected by service customers. So, the proposed model can be used for different types of services. Although, the proposed method presents high level of reliability and usability, issues like risk management, and trust mechanisms in the relationship between service providers and service customers are not considered by the authors.

2.5.2 Distributed Systems Performance Models

Kalepu et al. [29] propose a QoS-based attribute model to define non-functional metrics of distributed services. Availability, reliability, throughput, and cost attributes are used in their work to define the performance of resources of a given service provider. Two approaches of resources are used to calculate the final value of reputation. The first resource is the local rating record. Ratings of services which were invoked by local customer are stored in this record. In the second resource, global ratings of all services that executed on resources of given service provider are stored. Although, Kalepu et al. discuss the need to use SLA parameters to calculate the value of performance metrics, they do not explain how these parameters can be linked to the local and global resources of the rating system. In [30], Yeom et al. provide a monitoring methodology for the performance parameters of service. The proposed methodology uses the broker monitoring systems to evaluate the performance of resources of the service provider. Collected performance metrics data are not maintained in the service consumer database. This method incurs low cost in implementing the

measurement architecture, but has more risk in terms of privacy and availability of data; moreover, the security risks cannot be easily controlled, especially in the case of multi-tenant distributed systems. Kim et al. in [31] analyse the quality factors of performance level of services and propose a methodology to assign priorities message processing of distributed based on the quality factors of services. The process of determining the priorities in their framework is a dynamic process in different domain of services. They claim that their framework satisfy the agreement of service level in Web Services. However, the validation methodology of the proposed work lacks a clear definition of the evaluation criteria and the set-up of the experiment which yielded the claimed results. The work proposed by Guster et al. in [32] provide an evaluation methodology of distributed parallel processing. In the proposed method, authors use a parallel virtual machine (PVM) and real hosting servers to compare the results of their experiments. The efficiency of the evaluation method performed better in PVM for the processing time. On the real server environment, conducted experiments present better performance in the communication time. This work's method of evaluation does not include the implementation processes and it does not provide a clear explanation of the experiment results.

2.5.3 Cloud Computing Performance Models

Several studies on the scalability of virtual machines already exist [33]. Most of these studies considered the measurement of performance metrics on the local machines. The background loads of tested machines are controlled to compare the results of performance with a different scale of loads. Evangelinos and Hill [34] evaluated the performance of Amazon EC2 to host High Performance Computing (HPC). They use a 32-bit architecture for only two types of Amazon instances. Jureta, and Herssens [35] propose a model called QVDP which has three functions: specifying the quality level, determining the dependency value, and ranking the quality priority. These functions consider the quality of services from the customers' perspective. However, the performance issues related to cloud resources are not discussed and details are missing regarding the correlation of the quality model with the costing model of services. Cherkasova and Gardner [36] use a performance benchmark to analyse the scalability of disk storage and CPU capacity with Xen Virtual Machine Monitors. They measure the performance parameters of a visualization infrastructure that is already deployed in most data centres. But they do not measure the scalability of cloud providers using the visualization resources. However, my proposed work profiles the performance of

virtualization resources that are already running on the infrastructure of existing cloud providers.

The shortcomings of the proposed works for the above performance models are as follows:

1. The above proposed models for evaluation of the virtualization services focus on how to measure the performance of virtual machines on local experiments. However, the techniques for measuring the actual resources of cloud providers need more effort to ensure some level of trust between service providers and customers.
2. Most of the proposed work on performance evaluation does not allow service customers to specify the parameters of the performance metrics. In cloud computing, service customers need more flexibility and a dynamic approach to modify the parameters of performance metrics in order to solve the problem of dynamic changes of service requirements and business models of customers.
3. The experiments conducted for the above proposed models do not specify the benchmarks for the performance evaluation.
4. In cloud computing architecture, the relationship between performance monitoring and costing metrics is very important. The proposed model does not link the results of performance monitory with the actual cost metrics of services. So, service customers are not able to build the trust relationship with service providers without having a real cost model of services.

2.6 TRUST AND REPUTATION IN CLOUD COMPUTING

Trust concepts have been used in many areas such as economics, law, commerce, and information technology. Many researchers have investigated the various challenges to trust management. The amount of literature relating to this topic is increasing as researchers continue to discuss different issues and propose innovative models to solve the problems that arise when two parties need to establish a business connection between them. A variety of meanings has been attached to the term ‘trust’ in multiple dimensions. So, some of the literature in this area is confusing when the use of the trust concept is used in projects, but with different definitions [37].

Most of the definitions of trust in the literature are not complete, formal definitions. When the notion of trust appears in the literature, it is often without a formal definition. For instance, Deutch and Gambetta discuss the theoretical background and provide a basic definition of the trust concept for use in the real world [38]. An overview of trust and reputation definitions from the existing literatures presented by Hussain et al. [37] show that the current notions of trust and reputation need to be formally defined. Many researchers use the definition presented by Dasgupta [39] who defines trust as: "the expectation of one person about the actions of others that affects the first person's choice, when an action must be taken before the actions of others are known". Deutsch [40] states that: "trusting behavior occurs when a person encounters a situation where she perceives an ambiguous path. The result of following the path can be good or bad and the occurrence of the good or bad result is contingent on the action of another person" [37]. Another definition often cited in the literature is that given by Gambetta [41] "trust (or, symmetrically, distrust) is a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both before he can monitor such action (or independently of his capacity ever to be able to monitor it) and in a context in which it affects his own action".

Reputation mechanisms are used for large-scale open systems. In general, reputation can be regarded as the public's opinion about the object, character, or standing of entity such as reliability, capability, and usability. Users can provide ratings about a person, a product, an agent, or a service. Mui et al.[42] state that reputation is "a perception that an agent creates through past actions about its intentions and norms". Another definition presented by Abdul-Rahman et al.[43] is: "a reputation is an expectation about an agent's behavior based on information about or observations of its past behavior".

Reputation mechanisms are used in e-market systems (e.g. Amazon, e-Bay) to secure the transactions of all users in a centralized architecture. Novel models of reputation and trust have been developed in e-market places to provide reliable services of security since traditional solutions to security issues do not adequately protect providers and services consumers [44]. The most important aspect of these models is the information relating to past behaviours of users. This information is used to calculate and present the reputation of those users in terms of availability, reliability, and security. As a centralized architecture of online reputation models, e-Bay and Amazon exemplify this approach. Their systems are implemented based on a centralized rating model so that customers and sellers can rate each

other using numerical ratings or feedback comments. Users can obtain a reputation profile for a given user to decide whether or not to proceed with a transaction with this user. For example, e-Bay uses 1, 0, -1 scales which means positive, neutral, and negative feedback respectively. Users use these scales to rate business partners based on past behaviours. The feedback from users is stored in a central system and the reputation score is computed regularly as cumulative results of user ratings [45]. The problem with this mechanism is that users with high scores for reputation can cheat other users in a few transactions even though they receive negative feedback, because these users still gain positive ratings from other customers. Also, this model cannot guarantee the consistent delivery of all services from one user. This model employs a centralized architecture; therefore, all services and reputation information have a single point of failure.

Unlike the centralized architecture of service discovery, the peer-to-peer model does not use a single point to manage and store descriptions of services and reputation information. Vu et al. [46] propose peer-to-peer Web Service discovery that uses QoS and users' feedback to rank and select services. QoS data of services and reputation rates from consumers are stored in multi-peers in peer-to-peer systems. Monitoring agents are used to prevent cheating by users and providers. Trusted agents monitor and provide reports of services to a UDDI peer and based on this information, services are evaluated and ranked. However, monitoring reports differ from peer to peer, because each peer uses different criteria when providing feedback about services.

Dellarocas[47] proposed a model which detects and excludes any highly unfair ratings. In this approach, two important classes of reputation system frauds are addressed: (i) the users who are providing unfairly high ratings or unfairly low ratings for sellers, (ii) sellers who hide behind their good reputation in order to provide a service with low quality to different users on some occasions. To avoid the unfairly low ratings, Dellarocas uses controlled anonymity and cluster filtering methods. A collaborative filtering scheme is used to calculate an unbiased personalized reputation score. Using this method, groups of buyers who give similar ratings are grouped into two classes: upper and lower. The final reputation score is calculated using the lower classes only.

Yu and Singh [48] proposed a reputation system based on the Dempster-Shafer theory of evidence [49]. The proposed approach focused on detecting and protecting users against spurious ratings. Their method involves the use of a Weighted Majority Algorithm in order to

distinguish the local belief from the total belief. Local belief is derived from direct interaction and can be transferred to other users. Total belief is a combination of local belief and external recommendations received from any user.

Elnaffar in [50] proposes a reputation-based architecture for communities. In this architecture, UDDI is extended with user and provider agents connected to the reputation system. Elnaffar presents a reputation architecture to solve the problem of selecting from a community of in order for the consumer to discover a service, and the provider to publish a service. The community's architecture of is presented with details of functional operations to manage services in a particular service domain. Reputation, from the perspective of users and from the perspective of providers, has been discussed in this paper by the authors. In demand, selectivity, and market share metrics are used in this model to support providers when they want to sign up to an interested community. In each community, there is a master Web Service and slave Web Services. The master Web Service has the responsibility of attracting Web Services to subscribe to the community. This model is not suitable when one community has only a low number of slave Web Services. In this case, there are only a limited number of interactions between users and providers. Therefore, a reputation ratings number is limited also. The community with few members takes a long time to acquire a high score for reputation. Another problem with such community architecture is that when the master Web Service fails, all services within that community will fail to connect with this model.

Another model has been proposed by Shaikh in [51] for reputation-based semantic service discovery. Different contexts are used to compute the reputation of services. These contexts are based on particular application domains, or particular types of users. A reputation manager service is used to collect ratings from consumers. A weight coefficient is used to determine the importance level of the type of user or type of application domain and based on this, a coefficient reputation score is computed. The reputation framework is implemented in three phases. First, a Matchmaker system matches available services to user requirements. Second, a service composer enhances the availability of services, by combining a number of services in order to provide the required functionality in case no service is available that meets user requirements. The composer system draws on two different sources of information. In the rule-based approach, the service composer matches the input and output of composition templates with services in UDDI. This process is repeated until a suitable

template has been retrieved; failing this, a message is sent that no services match the user's requirements. On the other hand, the chaining approach can be used to create a chain of services to fulfil the objective(s) of the user. Although this service composer provides availability of services, it adds extra time to the discovery process. So, users who consider response time as a high priority do not benefit from this approach. The last phase is that of reputation computation where a reputation score is computed based on one of the approaches including service retrieval, atomic or composite services. In each approach, there is a different reputation function to compute the reputation score.

Maximilien and Singh [52] propose a model with specialized agencies to aggregate reputation and endorsement information. The reputation result is based on a collection of consumer ratings of services based on consumer preferences. Trustworthy providers and consumers can endorse new services in order to establish their reputation. On the other hand, an external advertisement agency is used to present new services as trusted services to be consumed by consumers. However, endorsement introduces providers as preferable agents, but it does not provide enough information about the performance of service. Moreover, this work does not specify how to compute a score for reputation.

The Classification of Trust and Reputation Models

The proposed models for trust and reputation systems can be analysed and classified from different perspectives. In this section, a set of classification dimensions is used to classify the computational models of the trust and reputation systems. The special characteristics of these models are considered in order to make the classification more clear in my study.

Following are the dimensions of the classification scheme:

(1) Computational algorithm

Trust models can be classified according to whether they are summation, average, fuzzy, or statistical computational methods. In the summation models, such as the system of e-Bay.com [45], the values of the feedback are added together to present the final value of reputation. The averaging method is used of Amazon.com. In this method, the number of transactions is included in the calculation process to make the reputation more accurate and clear to all users of the system. Fuzzy models of trust and reputation use fuzzy logic concepts. Using fuzzy rules, the trustworthiness of

users can be described in detail in terms of multi-domains of services. Models which have been developed based on the statistic calculation method use probabilities to estimate the trust values of users based on past interaction behaviours.

(2) Subjective or objective trust

The quality of services in some cases can be measured. In this situation, the trustworthiness of services is an objective trust. An example of objective trust is the service with a price parameter. Users can monitor the price and measure this parameter to give an objective feedback about services. Another type of trust is subjective trust. In this situation, users can not measure objectively the quality of services. They provide feedback about the service based on their opinions. For example, music download users use their individual opinion about music. This opinion reflects subjective user feedback but does not necessarily indicate the real quality of services. Models of trust and reputation can be classified according to these concepts of trust.

(3) Information source

Trust and reputation systems can be classified based on the source of the information retrieved to build the trust value. Local trust value is usually calculated using the local repository of the information about the direct interactions. On the other hand, external resources are used with models which calculate the trust value based on both local information and global reputation.

(4) Discrete or continuous feedback

Consumers of provide their feedback about the quality of services using two types of information. Discrete feedback is used to present the opinions of users as qualitative concepts. For instance, services may be evaluated as being very good, good, normal, or poor. On the other hand, the same services can be evaluated using continuous feedback in order to present the users' opinions within a limited period. Periods such as [-1, 1] and [0,1] can be used to evaluate services with this quantitative method of feedback. The trust and reputation models involved in the literature review are classified based on the dimensions above and a summary of this classification is presented in Table 2.2.

Table 2.2: A summary of trust and reputation models classification

Models	Computational algorithm	Subjective or objective trust	Information source	Discrete or continuous feedback
e-Bay.com [45]	Summation	Subjective	Local information	Discrete
Amazon.com [53]	Average	Subjective	Local information	Discrete
L.Vu et al. [54]	Average	Objective	External trusted agents	Continuous
C. Dellarocas [55]	Statistical	Subjective	Local and global information	Continuous
Yu & Singh [48]	Summation	Subjective and objective	Local and global information	Discrete
Elnaffar [50]	Average	Subjective	Local and global information	Discrete
Shaikh [51]	Fuzzy	Objective	Local information	Continuous
Maximilien & Singh [52]	Summation	Objective	Global information	Discrete

There is much in the existing literature on trust and reputation systems. However, it is impractical to present all of these works. However, from the above discussion, it is evident that the proposed works on trust and reputation management systems are designed mainly to

enhance the security of the traditional Web Services. In cloud computing, the execution of services has changed so as to be completely independent of consumer's infrastructure. Additionally, the price model for using cloud provider data resources is not same as the price of the traditional Web Services model. So, cloud computing lacks new trust and reputation approaches which can be integrated with the new technology and dynamic model of pricing. My proposed model will present a novel architecture of trust for cloud computing which take into account the above mentioned aspects of cloud computing. This architecture will use an SLA and business activities monitoring method to guarantee the quality of cloud services.

In the existing body of literature on cloud computing, there is no framework by which a cloud service consumer can make an intelligent trust-based decision regarding service selection from a service provider. Given the potential growth of cloud computing and the business implications, it is very important to have such architecture in place. In my research, I propose an architecture which is primarily SLA-based for selecting a given cloud service provider.

The Shortcomings of the Proposed Works for Trust and Reputation

1. The proposed models mainly use feedback from the users of services. Because ratings from service users present different subjective views of service performance, real data such as the results from SLAs monitoring agents are very important to reflect the actual level of trust and reputation of a service provider.
2. Many of the existing trust and reputation models are validated by simulation experiments. Simulation tools are not enough to evaluate the trustworthy of service providers under various real scenarios. There is a need to move from validation of trust and reputation models from simulation environments to real-world operational environments.
3. From the above discussion, it is evident that the proposed works in trust and reputation management systems are designed mainly to enhance the security of the traditional Web Services. These models do not cater for the trust management requirements in cloud computing which have specialized requirements such as scalability, dynamic pricing feature and integrity. So, cloud architecture lacks a novel approach to enhance the security and improve the service performance by using the concept of trust and reputation. In this thesis, I aim to address this shortcoming.

2.7 CONCLUSION

In this chapter, I have reviewed the proposed architectures for cloud computing and I discussed the differences between cloud computing and other distributed services. The discussion of cloud definitions, taxonomy, and shortcomings of the existing definitions of cloud computing are presented to provide a brief overview of the main concepts of the new paradigm of cloud computing. Then, I discussed the problems of existing schemes of service pricing. Then, I analysed the existing proposed framework of service level agreements in the context of Web Services, grid computing, and cloud computing. I identified and discussed the main shortcomings of the existing frameworks of SLAs. Then, I discussed the existing models of performance measurements for SOA, distributed services, and cloud computing followed by the list of the main shortcomings of these models. Finally, I presented the current approaches to the issues of trust and reputation in distributed environments.

In the next chapter, after the analysis of state of art, I define the problem of this thesis and propose a methodology for solving this problem.

REFERENCES

[1] H. Katzan Jr, *"On An Ontological View Of Cloud Computing"*, Journal of Service Science (JSS), vol. 3, pp. 1-6, 2011.

[2] D. Wyld, *"Moving to the cloud: An introduction to cloud computing in government",* Technical Report, IBM, Center for the Business of Government, Washington D.C, http://www.businessofgovernment.org/sites/default/files/CloudComputingReport.pdf, 2009, Accessed on December 6, 2010.

[3] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, *"A break in the clouds: towards a cloud definition"* ACM SIGCOMM Computer Communication Review, vol. 39, pp. 50-55, 2008.

[4] L. Wang, G. Von Laszewski, A. Younge, X. He, M. Kunze, J. Tao, and C. Fu, *"Cloud computing: a perspective study"*, New Generation Computing, vol. 28, pp. 137-146, 2010.

[5] Knorr, E., & Gruman, G, *"What cloud computing really means?",* InfoWorld, 2011, http://www.infoworld.com/article/08/04/07/15FEcloud-computing-reality_1.html, Accessed on March 2011.

[6] P. Mell and T. Grance, *"Draft nist working definition of cloud computing"*, http://csrc.nist.gov/groups/SNS/cloud-computing/index.html, Accessed on June 2009.

[7] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, *"Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility"*, Future Generation Computer Systems, vol. 25, pp. 599-616, 2009.

[8] R. Jennings, *"Cloud computing with the Windows Azure platform".* Wiley Publishing, 2009.

[9] C.N. Hoefer, and G. Karagiannis, *"Taxonomy of cloud computing services"*, IEEE Globecom 2010 Workshop on Enabling the Future Service-Oriented Internet, 2010.

[10] P. Laird, *"Different strokes for different folks: a taxonomy of cloud offerings"*, Enterprise cloud submit, INTEROP, 2009.

[11] B. P. Rimal, E. Choi, and I. Lumb, *"A taxonomy, survey, and issues of cloud computing ecosystems"*, Cloud Computing, pp. 21-46, 2010.

[12] D. Oliveira, F. A. Baião, and M. Mattoso, *"Towards a taxonomy for cloud computing from an e-science perspective"*, Cloud Computing, pp. 47-62, 2010.

[13] J. Joutsensalo, T. Hämäläinen, K. Luostarinen, and J. Siltanen, *"Adaptive scheduling method for maximizing revenue in flat pricing scenario"*, AEU-International Journal of Electronics and Communications, vol. 60, pp. 159-167, 2006.

[14] M. L. Honig and K. Steiglitz, *"Usage-Based Pricing of Packet Data Generated by a Heterogeneous User Population"*, INFOCOM '95, Fourteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Bringing Information to People. pp. 867-874 vol.2, 1995.

[15] Y. Hayel and B. Tuffin, *"An optimal congestion and cost-sharing pricing scheme for multiclass services"*, Mathematical Methods of Operations Research, vol. 64, pp. 445-465, 2006.

[16] H. Ludwig, A. Keller, A. Dan, R. P. King, and R. Franck, *"Web service level agreement (WSLA) language specification"*, IBM Corporation, 2003.

[17] Tosic. V, *"WSOL Version 1.2",* Technical Report, SCE-04-11, Department of Systems and Computer Engineering, Carleton University, 2004.

[18] A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, J. Pruyne, J. Rofrano, S. Tuecke, and M. Xu, *" agreement specification (WS-Agreement)"*, Open Grid Forum, 2007.

[19] A. Sahai, S. Graupner, V. Machiraju, and A. van Moorsel, *"Specifying and monitoring guarantees in commercial grids through SLA",* Cluster Computing and the Grid, Proceedings. CCGrid 2003. 3rd IEEE/ACM International Symposium, pp. 292–299, 2003.

[20] A. Leff, J. T. Rayfield, and D. M. Dias, *"Service-level agreements and commercial grids"*, IEEE Internet Computing, vol. 7, pp. 44-50, 2003.

[21] H. Keung, J. Dyson, S. Jarvis and G. Nudd, *"Self-adaptive and self-optimising resource monitoring for dynamic grid environments"*, 15th International Workshop on Database and Expert Systems Applications (DEXA'04), pp. 689-693, 2004.

[22] J. Padgett, K. Djemame, and P. Dew, *"Predictive adaptation for service level agreements on the grid"*, International Journal of Simulation: Systems, Science and Technology, vol. 7, pp. 29-42, 2006.

[23] V. Stantchev and C. Schröpfer, "Negotiating and enforcing qos and slas in grid and cloud computing," Advances in Grid and Pervasive Computing, pp. 25-35, 2009.

[24] C. H. Wen, H. R. Shiau, C. Y. Wang, and S. Y. Wang, *"A SLA-based dynamically integrating services Saas framework"*, Frontier Computing. Theory, Technologies and Applications, International Conference, pp. 306-311, 2010.

[25] S. Kounev and A. Buchmann, *"Performance Modeling and Evaluation of Large-Scale J2EE Applications"*, 29th International Conference of the Computer Measurement Group (CMC) on Resource Management and Performance Evaluation of Enterprise Computing Systems, pp. 273-284, 2003.

[26] S. Chandrasekaran, J. Miller, G. Silver, I. Arpinar and A. Sheth, *"Composition, performance analysis and simulation of ",* Electronic Markets, vol. 13/2 , pp. 8–30. 2007.

[27] D. Rud, et al. "*Performance modeling of WS-BPEL-based web service compositions*", IEEE Services Computing Workshops (SCW'06), pp. 140-147, 2006.

[28] D. A. Menascé, E. Casalicchio, and V. Dubey, *"A heuristic approach to optimal service selection in service oriented architectures"*, Proceedings of the 7th international workshop on Software and performance, pp. 13-24, 2008.

[29] S. Kalepu, S. Krishnaswamy, and S. W. Loke, *"Verity: a QoS metric for selecting and providers"*, First Quality Workshop, Rome, Italy, pp. 131-139. 2003.

[30] G. Yeom and D. Min, *"Design and implementation of QoS broker"* Proceeding of The International Conference on Next Generation Practices (NWeSP 2005), pp. 459-461, 2005.

[31] Kim, D., Lee, S., Han, S., Abraham, A, *"Improving Performance Using Priority Allocation Method"*, Proc. of International Conference on Next Generation Practices, pp. 201–206, Los Alamitos, 2005.

[32] D. Guster, A. Al-Hamamah, P. Safonov, and E. Bachman, *"Computing and network Performance of a distributed parallel processing environment using MPI and PVM communication methods"*, Journal of Computing Sciences in Colleges, vol. 18, pp. 246-253, 2003.

[33] M. A. Vouk, E. Sills, and P. Dreher, *"Integration of High-Performance Computing into Cloud Computing Services"*, Handbook of Cloud Computing, pp. 255-276, 2010.

[34] C. H. Constantinos Evangelinos, *"Cloud Computing for parallel Scientific HPC Applications: Feasibility of Running Coupled Atmosphere-Ocean Climate Models on Amazon's EC2"*, The First Workshop on Cloud Computing and its Applications (CCA'08), vol. 2, pp. 2-34, 2008.

[35] I. Jureta, C. Herssens, and S. Faulkner, *"A comprehensive quality model for service-oriented systems"*, Software Quality Journal, vol. 17, pp. 65-98, 2009.

[36] L. Cherkasova and R. Gardner, *"Measuring CPU overhead for I/O processing in the Xen virtual machine monitor"*, Proceedings of The Annual Conference on USENIX Annual Technical Conference, pp. 24-24, Anaheim, CA, 2005.

[37] F. K. Hussain and E. Chang, *"An overview of the interpretations of trust and reputation",* The Third Advanced International Conference on Telecommunications, Mauritius, pp. 30, 2007.

[38] A. C. Squicciarini, E. Bertino, E. Ferrari, I. Ray, and D. I. e Comunicazione, *"Achieving privacy in trust negotiations with an ontology-based approach"* IEEE Transactions on Dependable and Secure Computing, vol. 3, pp. 13-30, 2006.

[39] A. Dasgupta and A. Prat, "Reputation and asset prices: A theory of information cascades and systematic mispricing," *Manuscript, London School of Economics,* 2005.

[40] M. Deutsch, Distributive justice: *"A social-psychological perspective",* Yale University Press NewHaven, CT, 1985.

[41] D. Gambetta, *"Trust: Making and breaking cooperative relations",* Basil Blackwell New York, 1990.

[42] L. Mui, M. Mohtashemi, and A. Halberstadt, *"A computational model of trust and reputation",* Proceedings of the 35th Hawaii International Conference on System Science, pp. 7-10, 2002.

[43] A. Abdul-Rahman and S. Hailes, *"Supporting trust in virtual communities",* 33rd Ann, Hawaii Int"l Conf. System Sciences (HICSS-33), System Sciences, pp. 9, 2000.

[44] T. Grandison and M. Sloman, *"A survey of trust in internet applications",* IEEE Communications Surveys and Tutorials, vol. 3, pp. 2-16, 2000.

[45] P. Resnick and R. Zeckhauser, "*Trust among strangers in Internet transactions: empirical analysis of eBay's reputation syste,",* Advances in Applied Microeconomics: A Research Annual, vol. 11, pp. 127-157, 2002.

[46] L. H. Vu, M. Hauswirth, and K. Aberer, *"Towards p2p-based semantic web service discovery with qos support",* Proceeding of Workshop on Business Prpcesses and Services (BPS). Nancy, France, pp. 18-31, 2005.

[47] C. Dellarocas, *"Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior",* Proc. Second ACM Conf. Electronic Commerce, pp. 150-157, 2000.

[48] B. Yu, M. P. Singh, and K. Sycara, *"Developing trust in large-scale peer-to-peer systems",* Proceedings of First IEEE Symposium on Multi-Agent Security and Survivability, pp. 1-10, 2004.

[49] J. Gordon and E. H. Shortliffe, *"The Dempster-Shafer theory of evidence*", Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project, pp. 272-292, 1984.

[50] S. Elnaffar, Z. Maamar, H. Yahyaoui, J. Bentahar, and P. Thiran, *"Reputation of communities of -preliminary investigation",* AINA'08 Workshop Proceedings, pp. 1603-1608, 2008.

[51] S. Majithia, A. S. Ali, O. F. Rana, and D. W. Walker, *"Reputation-based semantic service discovery",* 13th IEEE Intl. Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), pp.297-302, Modena, Italy. 2004.

[52] E. M. Maximilien and M. P. Singh, *"Reputation and endorsement for ",* ACM SIGecom Exchanges, vol. 3, pp. 24-31, 2001.

[53] T. Grandison and M. Sloman, *"A survey of trust in internet applications",* Communications Surveys & Tutorials, IEEE, vol. 3, pp. 2-16, 2000.

[54] L. H. Vu, M. Hauswirth, and K. Aberer, *"Towards p2p-based semantic web service discovery with qos support",* BPM Workshops, pp. 18-31, 2005.

[55] C. Dellarocas, *"Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior",* ACM Conference on Electronic Commerce, p. 150-157, 2000.

CHAPTER 3 – PROBLEM DEFINITION

3.1 INTRODUCTION

In this chapter, I define the research problem addressed in this thesis. I start with definitions of terms and concepts that will be used in this thesis and for defining the research problem and how I apply them. Then, I formally present the problem that this thesis is addressing. Finally, I discuss the methodology which is used to address the research problem.

3.2 KEY CONCEPTS

This section provides brief definitions of the concepts and terminologies which my thesis uses to define the research problem.

3.2.1 Cloud computing

I adopt the definition of cloud computing which, according to the NIST (National Institute of Standards and Technology), is "... a model for enabling convenient, on demand network access to a share pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management afford or service provider interaction" [1].

3.2.2 Data

I define data as "the environment where the cloud service operates". This environment is owned by the service provider who has the responsibility to install, operate, and provide security and privacy solutions to save users' data in a protected situation.

3.2.3 Cloud provider

I define cloud provider as "an entity who builds and operates the cloud service architecture in order to offer a public or private service model of cloud computing".

3.2.4 Cloud user

I define cloud user as "the consumer who rents and pays for the use of a cloud service for a specific period of time". The cloud user is not responsible for assuring the availability or security of cloud infrastructure.

3.2.5 Infrastructure as a Service (IaaS)

I define IaaS as “a type of cloud service which provides resources for end users as a virtualized infrastructure”. Resources may be CPU, network, or storage technology. Cloud users are normally managed by assigning access controls rules and operating rules in order to use resources.

3.2.6 Platform as a Service (PaaS)

I define PaaS as “a way of delivering a computing platform and development tools as a service”. PaaS provides all the resources that are required to build, test, and deliver online applications without the need to pay large amounts of money in order to manage and update development platforms.

3.2.7 Software as a Service (SaaS)

I define SaaS as “a type of cloud service, whereby software applications are provided as a service rather than as ready-to-use software packages”.

3.2.8 Database as a Service (DaaS)

I define DaaS as “the database resources which can be offered via the Internet as a service”. With this type of service, multi-tenants can use the same physical resources to store their sensitive data.

3.2.9 Virtualization

Virtualization concept in the IT domain refers to any resources or technologies that are developed in a virtual rather than a real way. Virtual machines in cloud computing are virtualized machines that provide service as do actual machines. The main purpose of developing virtual machines is to obtain the advantage of scalability and to manage the allocation of cloud computing resources.

3.2.10 Boot time

I define the boot time of a virtual machine as “the time from when the user turns on the virtual machine to the time that it is ready to be used”.

3.2.11 Response time

I define the response time as “the time lapse between submitting a request and receiving the response to the request”.

3.2.12 Scalability

I define scalability as “the ability of a service to work properly when the system load either increases or decreases”.

3.2.13 Objective trust

The quality of services in some cases can be measured. In this situation, the trustworthiness of services is an objective trust. An example of objective trust is the service with a price parameter. Users can monitor the price and measure this parameter to give an objective feedback about services.

3.2.14 Subjective trust

The other type of trust is subjective trust. In this situation, users cannot objectively measure the quality of services. They provide the feedback on the service based on their opinions. For example, music download users use their individual opinion about music. This opinion indicates subjective user feedback but does not necessarily reflect the real quality of services. Models of trust and reputation can be classified according to these concepts of trust.

3.2.15 Service level agreements (SLAs)

Andrieux et al. [2] define a service level agreement as an online agreement that describes the agreed service, service level parameters, and actions when the service is not provided according to the required level of quality or performance [2]. SLA is one of the most important guarantees for the continuity of a business that uses online services as a part of business processes.

3.2.16 SLA metrics

I define SLA metrics as "metrics used to present the actual level of quality and performance of service objectives". These metrics can be retrieved directly from cloud resources such as CPU capacity or collected from third party agents such as users' feedback.

3.2.17 Quality of service (QoS)

I define QoS in the cloud computing environment as "the level of the user's acceptance of the cloud service provided by the service provider".

3.2.18 Negotiation

I define negotiation as "the activities between user and cloud provider before establishing a business relationship". Negotiation processes may include negotiation about IT objectives, business metrics, or management rules to obtain the guarantee of service.

3.2.19 Costing model

I define the costing model as a "mechanism used to determine the total cost over the whole period of the cloud service usage".

3.2.20 Service level objectives

I define service level objectives as "the level of measurable characteristics of cloud services". Examples of service level objectives include response time, boot time and availability.

3.3 DEFINITION

This section presents definitions of cloud computing, service level agreement, trust and reputation.

3.3.1 Cloud Computing

In the literature, various definitions of cloud computing have been proposed [1, 3, 4]. In this thesis, I adopted and considered the definition provided by the U.S. NIST (National Institute of Standards and Technology) that describes cloud computing as "... a model for enabling convenient, on demand network access to a share pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management afford or service provider interaction" [1]. In other words, cloud computing is a framework by means of which virtualized infrastructure resources are delivered as a service to customers by using a public network which is the Internet [5-7]. The cloud customers can range from big organizations, small business and developers to individual users. Examples of current cloud providers include: Amazon EC2 [8] (infrastructure cloud provider), Azure [9] from Microsoft (platform cloud provider), and for an application cloud provider, there is Google Docs [10]. In cloud computing, a virtualization technology is built on top of the infrastructure in order to optimize the use of resources and provide flexible solutions for users.

3.3.2 Service Level Agreements

A Service Level Agreement (SLA) [11-13] is a contract that describes the agreed service, service level parameters, guarantees, and actions and remedies for all cases of violation [2]. The SLA is very important as a contract between consumer and provider. The main purpose of SLAs is to clarify and formalize the agreements about service terms such as performance, availability and billing between the cloud customers and providers. It is important that the SLA include the obligations and the actions that will be taken in the event of any violation, with clearly shared semantics between each party involved in the online contract. The SLA is a legal format documenting the way that services will be delivered as well as providing a framework for service charges. Service providers use this foundation to optimize their use of

infrastructure to meet signed terms of services. Service consumers use the SLA to ensure the level of quality of service they need and to maintain acceptable business models for long-term provision of services. The following are the main requirements of the SLA:

1. The SLA format should clearly describe a service so that the service consumer can easily understand the operation of the services.
2. The SLA presents the agreed level of performance of service (from the perspective of both parties).
3. It defines ways by which the service parameters can be monitored and the format of monitoring reports.
4. It must specify the penalties when service requirements are not met.
5. The SLA presents the business metrics such as billing and stipulates the conditions under which this service can be terminated without any penalties being incurred.

In this research, the focus will be on the non-functional requirements of services such as availability, scalability and response time. Based on the more important non-functional requirements, the SLA parameters for each type of cloud service can be defined as:

1. Availability: in cloud computing, the most important criterion for quality of service is the availability of service. Availability is the probability that the cloud infrastructure or services will be up and running in the specific time of utilities of the service provided for in the SLA.
2. Scalability: cloud consumers pay for the service only as they use it. The cloud provider should facilitate the specific resources for ease of scaling up and down. With scalability, cloud consumers can maximize revenue and cloud providers are able to optimize resources effectively.
3. Resource reservation: this is not a method that is unique to each type of cloud service. For example, the storage service can be billed based on the time and size of the user's data. On the other hand, cloud CRM (Customer Relationship Management) may be billed based on the number of users. This research will include an investigation of the most suitable cost calculation method for these types of cloud services [14].

4. The configuration of service: in cloud computing, users deal with virtual machines and these VMs should be configured in a flexible manner to enable users to execute business processes with minimal need for managing the configuration.

5. Security and privacy: the critical data of a business must be stored and transferred via secure channels. If security features are not guaranteed by cloud providers, business organizations may spend too much on operating their own data centres rather than switching to cloud providers [15].

3.3.3 Concepts of Trust and Reputation in Cloud Computing

With the number of users of online systems increasing, trust and reputation issues have become the main obstacles preventing the users of these technologies from developing relationships between strangers and unknown services. Most traditional security solutions used for solving this problem cannot adequately meet the requirements of service users. Trust and reputation are essential components of any interaction with unknown providers of services in distributed systems. For instance, authentication and authorization are not enough to stop the malicious introduction of viruses and other malicious codes [16]. Also, by using a Public Key Infrastructure (PKI), users can change their identities and rejoin the system after they have been prevented from doing so using the first identity [17]. In fact, the problem remains of how service consumers can interact with service providers in distributed systems and invoke the most reliable service from an array of services. One solution to such problems is to use the trust and reputation methods to support the current solutions of security in online systems. This section provides definitions of trust and reputation concepts in the context of distributed services.

3.3.3.1 Trust Definition

Trust concepts have been used in many areas such as economics, law, commerce, and information technology. Many researchers have investigated the various challenges to trust management. The amount of literature relating to this topic is increasing as researchers continue to discuss different issues and propose innovative models to solve the problems that arise when two parties need to establish a business connection between them. A variety of meanings has been attached to the term ‘trust’ in multiple dimensions. Hence, some of the

literature in this area is confusing when different definitions of the trust concept are used in projects [18].

When the notion of trust appears in the literature, it is often without a formal definition. For instance, Deutch and Gambetta discuss the theoretical background and provide a basic definition of the trust concept for use in the real world [19]. An overview of trust and reputation definitions from the existing literature presented by Hussain et al. [18] shows that the current notions of trust and reputation need to be formally defined. Many researchers use the definition presented by Dasgupta[20] who defines trust as: "the expectation of one person about the actions of others that affects the first person's choice, when an action must be taken before the actions of others are known". Deutsch [21] states that: "trusting behaviour occurs when a person encounters a situation where s/he perceives an ambiguous path. The result of following the path can be good or bad and the occurrence of the good or bad result is contingent on the action of another person" [18]. Another definition often cited in the literature is that given by Gambetta [22]: "trust (or, symmetrically, distrust) is a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both before he can monitor such action (or independently of his capacity ever to be able to monitor it) and in a context in which it affects his own action".

3.3.3.2 Reputation Definition

Reputation mechanisms are used for large-scale open systems. In general, reputation is defined as the public's opinion about the object, character, or standing of an entity such as reliability, capability, and usability. Users can provide ratings about a person, a product, an agent, or a service. Mui et al. [23] state that reputation is "a perception that an agent creates through past actions about its intentions and norms". Another definition presented by Abdul-Rahman et al. [24] is: "a reputation is an expectation about an agent's behaviour based on information about or observations of its past behaviour".

3.4 PROBLEM OVERVIEW AND PROBLEM DEFINITION

In Chapter 1, I outlined the motivation and importance of trust and measurement of cloud service performance to ensure the continuity of business and to improve the reputation of the

service provider in the cloud computing environment. In this section, I present an overview of the main problem of this thesis. In next section, I discuss the research issues and challenges related to the main problem of building a trust model for the cloud computing environment.

In terms of trust, the challenge for cloud architecture is to implement cloud computing systems in such a way as to increase the trust level of cloud users and fulfil customer requirements. Users of cloud services may take legal action in the case of any violations of service level agreements. Also, cloud providers may face legal action and damage the reputation of their organizations if they violate the terms of the SLA.

Moreover, an increasing number of cloud providers with various levels of services pose scientific challenges when cloud customers want to select the optimal and good services from a large number of different services. In this case, cloud customers need to have a reliable trust model to help them select the most trusted and secure services from the different levels of cloud services on offer.

Furthermore, cloud technology deploys virtual computing to process and store the data of cloud users. This type of technology increases the risk of storing and processing the sensitive data of customers and sensitive organizations. Cloud users without a sufficient level of security and trust values of cloud providers, are not willing to connect to untrustworthy providers of cloud services. As discussed in Chapter 2, no solution has been proposed to integrate the idea of trust concepts with the models of cloud service selection. More effort is still needed before moving to cloud data centres and storing sensitive data in remote virtualized infrastructures.

In Chapter 2, I stated that trust models and performance evaluation approaches have received more attention in the research community. However, the proposed trust approaches focus on the use of subjective assessments provided by users of distributed services. In the case of cloud architecture, the dynamic nature of the allocation of cloud resources raises the need to develop novel frameworks of trust and performance evaluation. These frameworks should take into account the QoS aware based systems and integrate QoS evaluation methods with the trust solution for cloud computing. In my presentation of trust models in Chapter 2, it is clear that no models use the QoS evaluation method for cloud architecture to develop a reliable trust model that can provide an accurate methodology for cloud service selection.

The existing literature on trust and reputation focuses on investigating trust and reputation solutions for general domains of distributed services. So, the proposed models can be applied to different architectures of services. For instance, the proposed models can be modified with some changes to the peer-to-peer architecture, , and e-business services etc. To the best of my knowledge, the existing solutions of trust and reputation do not consider the new paradigm of cloud architecture. Trust and reputation in cloud computing lack dynamic and scalable models because cloud computing has unique features which are not found in other types of distributed services. For example, in cloud computing, trust and reputation models should be developed to solve the problem of virtualization trust, dynamic model of cost, reputation of cloud providers, and type of service model of cloud computing. All these features have to be considered in every stage of the trust life cycle when developing trust-based models for cloud computing.

Unfortunately, the proposed methods for securing and protecting data on distributed architecture use either hardware trusted-based systems, or encrypted data-based models. The solutions that use trust-based systems lack scalability features. On the other hand, encrypted data-based models are still in the theoretical stage and need to be further developed in order to be applied to the cloud computing architecture. Moreover, since the resources of a virtualized infrastructure are under the control of cloud providers, cloud customers not only lack reliable solutions for trust management when dealing with cloud providers, but they are also concerned about the configurations and management activities of cloud data centres. In order to provide a generic solution for assessing the reputation of cloud providers and evaluating the trust of cloud resources, it is essential to combine the subjective concepts of trust and real-time measurement of cloud resources so as to provide a generic framework of trust management for the cloud community.

The lack of research into reviewing and analysing the current requirements of cloud customers in terms of security and trust management for cloud systems, adversely affects developers and distributed services providers who need to develop effective models for service provisioning and business in cloud-based investments. To the best of my knowledge, (and as discussed in Chapter 2) there is no study in the existing literature that has surveyed and reviewed methods of analysing the requirements of cloud users from the perspective of trust and reputation. Research conducted using a survey approach may provide significant insights into user's requirements and real problems regarding the security and trust issues of the cloud community.

Moreover, service consumers of cloud computing are willing to pay as they use, so an annual billing period or even monthly periods are not suitable for cloud computing. The cost calculation for resource reservation in cloud computing is not a unique method for each type of cloud service. For example, the Database as a Service (DaaS) can be billed based on the time and size of the user's data. On the other hand, cloud Software as a Service (SaaS) may be billed based on the number of users. Cloud architecture lacks an effective model that meets consumer requirements and provides an acceptable level of revenue for cloud providers. In cloud computing, a dynamic pricing scheme is very important to allow cloud providers to estimate the price of cloud services. Moreover, the dynamic pricing scheme can be used by cloud providers to optimize the total cost of cloud data centres and correlate the price of service with the revenue model of service. In the context of cloud computing, dynamic pricing methods in terms of cloud providers and cloud customers are missing from the existing literature. A dynamic pricing model for cloud computing must consider all the requirements of building and operating cloud data centres. Furthermore, a cloud pricing model must consider issues of service level agreements with cloud customers.

Many organizations and e-services-based companies need reliable methods to compare the cost of building in-house IT centres when moving their systems to a cloud platform and estimate the price of service to maximize the revenue. It is important that the parameters of cost metrics such as cost of developing software, maintenance, and hardware of in-house IT centres be clearly correlated with the dynamic pricing of cloud services. A dynamic pricing methodology will provide adequate estimating methods for decision makers who want to calculate the benefits and assess the risks of using cloud technology.

Additionally, the existing cloud platforms in the market of cloud services do not offer a clear guarantee or well-defined service level agreements (SLAs) to satisfy different interests of cloud customers who need to ensure the continuity if their business for the long term. Cloud users do not want just a certain percentage of guarantee of availability; they require a more specific definition of an SLA that can be used as an agreement about the required quality or level of services for different applications. Based on the model of cloud service delivery, various parameters of SLA metrics can be included in the structure of SLA for cloud services. For example, hardware parameters such as CPU capacity, response time, and boot time of virtual machines are important parameters for IaaS. These parameters may be considered to be the most important parameters for users who are willing to use cloud computing as an operating system to run their application with some level of quality of services. On the other

hand, users of PaaS, SaaS, and DaaS may ask for different types of SLA parameters (customized depending on the cloud service) that should be included in SLA agreements. To the best of my knowledge, no SLA structure has been proposed in the existing literature that takes into account the context of different models of cloud services and the dynamic issue of parameters of SLA for cloud users.

Additionally, cloud data centres are established in a complicated way. Numerous racks of physical servers interconnected with network switches, and the complexity of virtualization resources, produce scientific challenges to the development of a generic prediction model for estimating the values of SLA parameters. To solve these problems, measurements of low level resources in cloud architecture are very important to map the objectives in SLA with real values of performance of cloud resources.

Moreover, service level agreements have been proposed for different domains by various researchers. However, as I discussed in Chapter 2, no proposed work for SLAs in cloud computing in the existing literature takes into account the problem of selecting SLA metrics for different models of services. How to select appropriate measurement metrics of cloud services is a critical process for users of cloud computing. Many issues should be considered in the process of selecting SLA metrics. For example, choosing metrics for an SLA which accumulates a sizeable amount of data does not work for users who are seeking a low level of response time for their application. So, there is no unique way of choosing measurement metrics of cloud services, since each type of business or service model has a specific SLA structure to provide more accurate results or produce a high amount of revenue using the cloud business model.

Finally, to the best of my knowledge, in the context of cloud computing, there is no approach in the literature that can be applied to predict trust and reputation for cloud relationships between service providers and cloud customers that provides a costing model of cloud services and good guidelines for SLAs to cloud users.

This thesis presents complete definitions of service level agreements and takes into consideration the performance metrics and quality of service criteria of cloud services. The proposed definitions of SLA are provided for use by different types of cloud users. In order to help cloud users include the most relevant factors of performance and quality of cloud services in SLAs, this thesis also provides a methodology to assess and measure the most relevant metrics of performance in different cloud platforms. I do this to enable cloud users to

choose appropriate criteria for performance of cloud services when they begin negotiations with cloud providers. Finally, I develop a dynamic pricing scheme for cloud services to be used with an SLA framework for developing a trust model for cloud users. The trust model for cloud relationships provides a reliable means of selecting cloud services with the required level of privacy and security.

Based on the discussion of the literature in Chapter 2 and the above description of the research problem, the research problem can be defined as follows:

How can a cloud service user choose the most trustworthy and secure cloud service from among different services in the cloud platform market that takes into account the structure of SLAs, dynamic pricing, and accurate metrics for performance measurement of cloud resources?

3.5 RESEARCH ISSUES

The thesis identifies the following important research issues that must be addressed in order to solve the thesis problem. The research issues are:

1. Propose a dynamic pricing scheme: The service consumers using cloud computing are willing to pay as they use, so an annual billing period or even monthly periods are not suitable for cloud users. A cost calculation method for resource reservation must be correlated with the method of proposing the price of service in order to maximize cloud service profits and increase the customer demand. Cloud architecture lacks an effective model to satisfy consumer requirements and provide an acceptable level of revenue for cloud providers.

2. Develop SLA metrics for cloud platforms: The SLA parameters are specified by metrics. These metrics define how cloud service parameters can be measured and specify values of measurable parameters. In the cloud computing architecture, there are four types of services which providers can offer to consumers. These services are: Infrastructure as a Service (IaaS), Platform as a Service, Software as a Service, and Database as a Service. Any proposed SLA metrics for cloud computing should consider these four types of services. For each part of the SLA, the most important

parameters should be defined so that consumers can use it to create a reliable model of negotiation with the service provider.

3. Propose a methodology to define performance and measurement criteria: Performance evaluation is a very important factor in the open distributed systems. It is the main concern of all interactions between service providers and consumers in such a changing environment. The performance evaluation process may not be clear and easy for some users because it has vague and different subjective values. Hence, what is required is a description methodology to present the values of performance metrics in a clear way. Any solution aimed at solving this problem should take into account the dynamic nature of distributed services and deploy an effective methodology to calculate the final values of performance metrics

4. Propose a trust and reputation model for cloud service selection: In the existing body of literature on cloud computing, there is no framework whereby a cloud service consumer can make an intelligent trust-based decision regarding service selection from a service provider. Given the potential growth of cloud computing and the business implications, it is very important to have such architecture in place. The primary issues which are not investigated in the related literature on this topic are:

    i. the difficulties faced by cloud users when they want to sign online agreements with cloud providers; there is no clear and reliable method for selecting the most suitable parameters for the SLAs;

    ii. the lack of a proposed model to estimate the price of cloud services.

    Although trust and reputation systems have been widely proposed and implemented for various types of online services, no such models have been proposed for cloud computing; cloud users also need such systems in order to select the most trustworthy of the services already being offered by cloud providers.

5. Validate the proposed solutions: Researchers in most cases evaluate and validate the proposed frameworks for performance measurement of distributed services using local experiments or simulation methods. These methods sometimes do not provide accurate results that reflect the actual level of trust and performance of services. Real

experiments need to be conducted on the resources of existing cloud infrastructures in order to evaluate the real components of services. The following are the main criteria for the evaluation processes of the proposed model:

(1) Scalability: To evaluate the performance of the proposed model when dealing with both small and large numbers of users, this simulation is conducted with different types of services and the simulation results are compared with the current models evaluated in the design phase of this research. The main parameters in the scalability evaluation are:

   i. the performance of the computation of trust model;

   ii. the performance of the communication overhead; and

   iii. the capacity of data storage.

(2) Effectiveness of the proposed model

The objective of evaluating the effectiveness of the proposed model is to guarantee that the functionalities of the model match the trust requirements of cloud users.

In order to evaluate the above metrics for the proposed trust model, I conduct experiments on a well-known cloud provider (Amazon EC2) by implementing real cloud services for the purposes of my project.

3.6 RESEARCH METHODS

This section discusses the research methodology which is followed to ensure that my research processes and the proposed solutions are addressed using a systematic and scientifically-based method. Because this research involves a literature review, design, development, and validation methods to provide a scientific trust framework for the cloud community, I discuss and justify the research method chosen as the most suitable for the purposes of my study.

According to Denzin and Lincoln, research methods are skills and practices that provide guidelines to researchers for addressing research questions and the research process [25]. They discuss the scope of problems, the research resources and the rules for researchers when

they undertake research projects. Action research, case studies, literature reviews and laboratory experiments are examples of research methods. The use of more than one method of research is feasible when researchers consider the integration of all processes of a research project [26]. Since the aim of this research is to develop and validate a trust-based service level agreement framework in the cloud computing environment, the researcher intends to use the Science and Engineering based research approach. This approach is intended to develop a scientific solution to the identified problem by first developing a theoretical framework and later confirming it by using simulations and experiments to confirm its spirit of 'making something work'[9, 10, 27]. The output of such a research approach will lead to improving the approaches of either previous or current studies. There are various research methodologies in this research approach. The method chosen by the researcher is the Systems Development based approach which is a combination of various research steps, such as: a) envisaging a system; b) developing key concepts and theories; c) developing a conceptual framework; d) building the system; and e) testing and validating the system. Each of the these steps has various sub-steps that can be classified into three broad categories as shown in Figure 3.1.

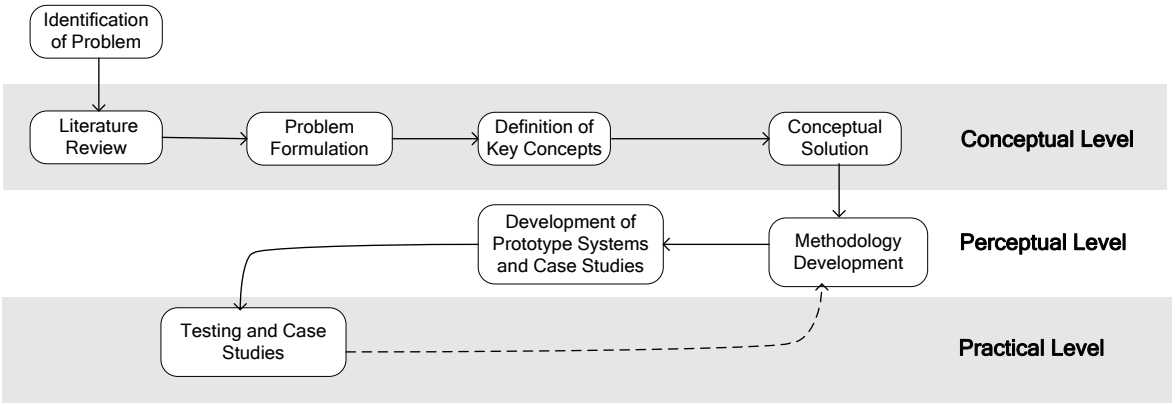


Figure 3.1**:** Different stages in System Development approach of Science and Engineering based research methodology

My research began with an extensive review of the literature related to the topics of cloud computing, security, trust and reputation. Also, the existing definitions and implementation of service level agreements, especially in distributed and cloud services, were evaluated. Most of the sources have been explored in order to understand the main area of cloud computing technology and the applications of trust and reputation systems in the online environment. To

understand the mechanisms of service level agreements in e-business, the existing literature in that area has been reviewed.

Design the Trust Model based on the SLA: The objective of this phase is to design an innovative model of trust and reputation based on the SLA approach. To assist cloud providers to define SLAs parameters, it is very important to correlate the cost parameters with the criteria for service quality. Hence, this research aims to develop a dynamic pricing scheme covering all tiers of cloud architecture beginning from hardware infrastructure to the end user interface. The aim of developing this dynamic pricing scheme is to help the decision makers in cloud data centres and service customers to establish a clear relationship in the cloud community based on the guarantee terms that correlate with the cost plan. Also, in this phase of the research, the general trust model for cloud providers will be designed, and different factors will be considered when calculating a fear value of trustworthiness of cloud providers. A weighting approach is designed to assign a correct value to each parameter of the proposed model. The number of violations, for example, is assigned more weight than is customers' feedback. Cloud users can use the output of the proposed model to select the most secure and reliable services from among a large number of offered services; this will help to reduce the risk in the event that users migrate to use the external resources of cloud computing.

Evaluation of the Proposed Model: In order to evaluate the effectiveness and robustness of the proposed model, and to learn if the model is efficient in helping users to select a reliable cloud provider, I conducted various experiments. These experiments tested the functions of the proposed model in order to be used in the cloud architecture.

3.7 CONCLUSION

In this chapter, I defined the main problem of this thesis. I began by defining the main concepts of the domain of cloud computing. Then, I provided definitions of SLAs, trust, and reputation. Next, I defined the main problem and its related issues which this thesis will address. The methodology used for solving the problem was discussed and I explained in depth the research approach and the main problem in relation to the existing literature.

REFERENCES

[1] P. Mell and T. Grance, *"Draft nist working definition of cloud computing"*,http://csrc.nist.gov/groups/SNS/cloud-computing/index.html, Accessed on June 2009.

[2] A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, J. Pruyne, J. Rofrano, S. Tuecke, and M. Xu, *" agreement specification (WS-Agreement)"*,Open Grid Forum, 2007.

[3] J. Napper and P. Bientinesi, *"Can cloud computing reach the top500?"*, UCHPC-MAW, Proceedings of the combined workshops on UnConventional High Performance Computing Workshop Plus Memory Access Workshop. New York, USA: ACM, pp. 17-20, 2009.

[4] Y. Chen, V. Paxson, and R. Katz, *"What's new about cloud computing security?"* Technical Report UCB/EECS-2010-5, UC Berkeley Department of EECS, 2010, http://www.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS-2010-5.pdf, Accessed on Jan 2010.

[5] R. Buyya, *"Market-Oriented Cloud Computing: Vision, Hype, and Reality of Delivering Computing as the 5th Utility"*, IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID'09), Shanghai, China, May, pp. 1, 2009.

[6] A. Marinos and G. Briscoe, *"Community Cloud Computing",* Lecture Notes in Computer Science, Vol. 5931/2009, pp. 472-484, 2009.

[7] P.T. Jaeger, T. Lin, and J.M. Grimes, *"Cloud computing and information policy: computing in a policy cloud"*, Journal of Information Technology & Politics, vol 5(3), pp. 269-283, 2008.

[8] E. Amazon, *"Amazon Elastic Compute Cloud (Amazon EC2)"*, http://aws.amazon.com/ec2/, Accessed on June 2010.

[9] Microsoft*, "Microsoft Azure",* Available: http://www.microsoft.com/windowsazure/, Accessed on June 2010.

[10] Google,"*Google Docs"*, Available: http://docs.google.com, Accessed on March 2010.

[11] M. Boniface, S. C. Phillips, A. Sanchez-Macian, and M. Surridge, *"Dynamic Service Provisioning Using GRIA SLAs"*, Service-Oriented Computing-ICSOC, International Workshops, Vol. 4907, Springer, pp. 56–67, 2007.

[12] G. Di Modica, O. Tomarchio, and L. Vita, *"Dynamic SLAs management in service oriented environments"*, The Journal of Systems & Software, vol. 82, pp. 759-771, 2009.

[13] A. Keller and H. Ludwig, *"The WSLA framework: Specifying and monitoring service level agreements for "*,Journal of Network and Systems Management, vol. 11, pp. 57-81, 2003.

[14] M. D. de Assunção, A. di Costanzo, and R. Buyya, *"Evaluating the cost-benefit of using cloud computing to extend the capacity of clusters"*, pp. 141-150, 2009.

[15] A. C. Barbosa, J. Sauvé, W. Cirne, and M. Carelli, *"Evaluating architectures for independently auditing service level agreements*", Future Generation Computer Systems, vol. 22, pp. 721-731, 2006.

[16] Y. L. Sun, Z. Han, W. Yu, and K. J. R. Liu, *"Attacks on trust evaluation in distributed networks",*Proc. of the 40 th Annual Conference on Information Science and Systems (CISS), pp. 1461-1466, Princeton, NJ, 2006.

[17] M. Lima, A. Dos Santos, and G. Pujolle, *"A survey of survivability in mobile ad hoc networks*", Communications Surveys & Tutorials, IEEE, vol. 11, pp. 66-77, 2009.

[18] F. K. Hussain and E. Chang, *"An overview of the interpretations of trust and reputation",*The Third Advanced International Conference on Telecommunications, Mauritius, pp. 30, 2007.

[19] A. C. Squicciarini, E. Bertino, E. Ferrari, and I. Ray, *"Achieving privacy in trust negotiations with an ontology-based approach*", IEEE Transactions on Dependable and Secure Computing, pp. 13-30, 2006.

[20] A. Dasgupta and A. Prat, *"Reputation and asset prices: A theory of information cascades and systematic mispricing"*,Manuscript, London School of Economics, 2005.

[21] M. Deutsch, "Distributive justice: *A social-psychological perspective*", Yale University Press NewHaven, CT:, 1985.

[22] D. Gambetta, *"Trust: Making and breaking cooperative relations",* Basil Blackwell New York, 1990.

[23] L. Mui, M. Mohtashemi, and A. Halberstadt, *"A computational model of trust and reputation",* Proceedings of the 35th Hawaii International Conference on System Science, pp. 7-10, 2002.

[24] A. Abdul-Rahman and S. Hailes, *"Supporting trust in virtual communities",* 33rd Ann, Hawaii Int"l Conf. System Sciences (HICSS-33), System Sciences, pp. 9, 2000.

[25] E. G. Guba, Y. S. Lincoln, and N. K. Denzin, *"Handbook of qualitative research"*, (2nd edition), Sage, Thousand Oaks, CA, 2000.

[26] J. Becker, D. Pfeiffer, and M. Rackers, *"Domain specific process modelling in public administrations,the picture approach",*Wimmer, M.A, Scholl, J, Gronlund, A, (eds), EGOV. LNCS, vol. 4656, pp. 68–79, Springer, Heidelberg, 2007.

[27] A. R. Hevner, S. T. March, J. Park, and S. Ram, *"Design science in information systems research",Mis Quarterly,* vol. 28, pp. 75-105, 2004.

CHAPTER 4 - SOLUTION OVERVIEW

4.1 INTRODUCTION

As I discussed in Chapter 2, several works in the existing literature attempt to solve the problem of trust and reputation in various domains of distributed services. Different techniques are used to enhance the security level of services. However, as mentioned and discussed in Chapter 2, the issues of a dynamic pricing scheme for cloud computing, defining service level agreements, measurement of performance, and modelling trust and reputation in the domain of cloud computing still remains unsolved. In Chapter 3, I discussed the research issues related to pricing schemes, designing SLAs, performance measurement, and trust and reputation for cloud computing. In this chapter, I provide an overview of the solutions for the research issues listed in Chapter 3. In Section 4.2, I present an overview of the solution for modelling service pricing of cloud computing. Section 4.3 sets out the solution for defining SLAs for each of the four main types of cloud services. In Section 4.4, I explain my solution to modelling the method of performance measurement of cloud services. In Section 4.5, I present an overview of the solution for modelling the trust and reputation for cloud services. Finally, I conclude the chapter in Section 4.6.

4.2 OVERVIEW OF THE SOLUTION FOR DYNAMIC PRICING SCHEME FOR CLOUD SERVICES

Dynamic pricing mechanism of cloud services is responsible for measuring the actual usage of resources and billing the cloud users for the consumption of resources. The requests are allocated to the proper resources and when this is done, the cost of doing so is charged to the customer [3]. Taking into account the historical usage information or profile of users would assist in the dynamic pricing method. This information can be retained and utilized by scalable admission control. Information like this helps to improve the resource allocation decisions and the corresponding billing of the users. In this part of my thesis, I discuss the different schemes that can be applied for pricing services that are being offered by online providers. Then, I present a dynamic pricing scheme for cloud services. Finally, I analyse the results and evaluate the proposed scheme.

To improve the way of pricing cloud services, I propose a dynamic pricing scheme for cloud service customers and cloud service providers. In the proposed scheme, a cloud market agent is used to provide the matching process and negotiation about service level objectives for cloud users. In this scheme, pricing functions are proposed to control the cost level from the perspective of cloud customers and to control the resource allocation and maximize the revenue for the cloud service providers. I discuss the problem of how to announce the service price when the demand of service and the resources are not stable in the cloud computing market. I use weighting parameters λ and σ to control the increasing and decreasing amount of pricing when the cloud market is high and then low. Also, for cloud customers, I propose a methodology to estimate the quality of cloud services and propose a price based on the quality level and time slot for conducting the cloud service. The conducted simulations show the proposed dynamic pricing scheme for cloud computing provides scientific results of improving the revenue for cloud service providers and helps cloud customers to rent a high quality of cloud services. The detailed solution for my dynamic pricing scheme and the underpinning mathematical models are presented in Chapter 5.

4.3 OVERVIEW OF THE SOLUTION FOR SERVICE LEVEL AGREEMENTS FOR CLOUD SERVICES

I present a methodology for answering the questions of "How can the performance of cloud services be measured and what are the performance metrics that can most affect the SLAs of different types of cloud services?

Cloud services are becoming popular in terms of distributed technology because they allow cloud users to rent well-specified resources of computing, network, and storage infrastructure. Users pay for their usage of services without needing to spend massive amounts for integration, maintenance, or management of the IT infrastructure. This creates the need for a reliable measurement methodology of this new paradigm of services. To solve the problem of linking service objectives with a pricing method for cloud services, I develop performance metrics to measure and compare the performance of the resources of virtualization on the cloud data centres. First, I discuss the need for a reliable and clear methodology to use for service level agreements of cloud services for the four broad types of cloud services (IaaS, PaaS, SaaS and DaaS). Second, I develop a specialized suite of metrics

and propose a suitable methodology to correlate the performance metrics with service level objectives for each type of cloud service.

The primary shortcoming of the existing approaches to SLAs in the online domain is that they do not provide dynamic negotiation, and various types of cloud consumers need a different structure of implementation of SLAs to integrate their own business rules with the guarantees that are presented in the targeted SLA. To solve such problems, I propose a basic architecture for developing the service level agreement contract between service consumers and other parties such as service providers and external agents. Two main categories of SLA metrics are presented. Performance metrics show the measurements of performance parameters in cloud computing data centres such as response time and CPU capacity. The other metrics are business metrics; the main measurements of business-related aspects presented by this type of metrics include such things as service costs and billing methods.

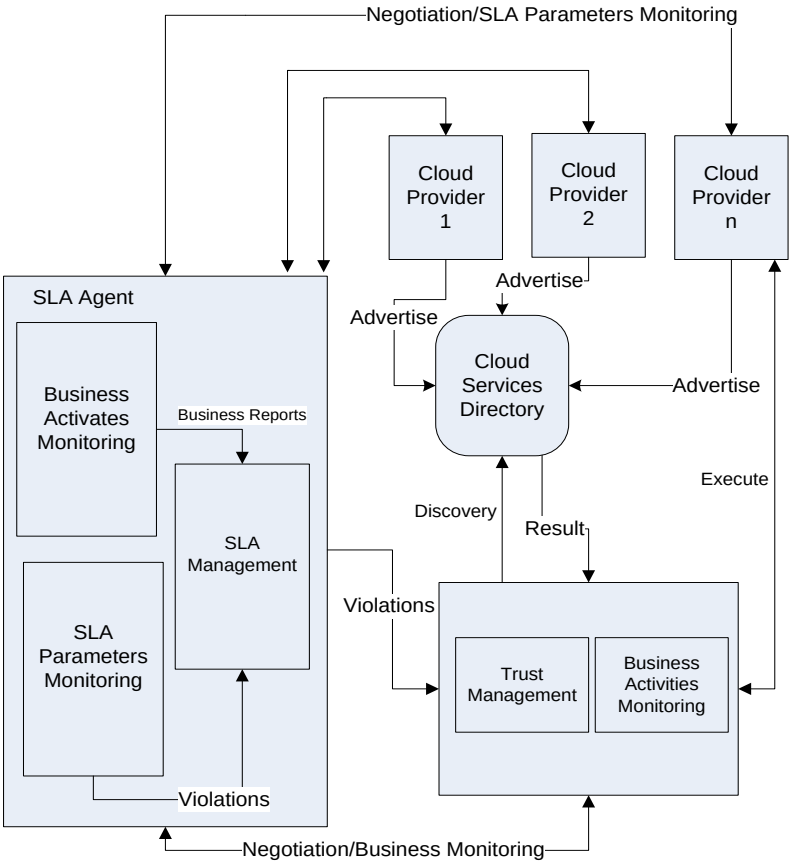


Figure 4.1: Conceptual SLA Framework for Cloud Computing

In my proposed framework (figure 4.1), the SLA parameters are specified by metrics for each of the different types of cloud services. These metrics define how cloud service parameters can be measured and specify values of measurable parameters. In the cloud computing architecture, there are four types of services which providers can present to consumers. These services are: Infrastructure as a Service (IaaS), Platform as a Service, Software as a Service, and Database as a Service. The proposed SLA metrics for cloud computing consider these four types of services. For each part of the SLA, I define the most important parameters that consumers can use for the corresponding service to create a reliable model of negotiation with this service provider.

In this part of my thesis, I present the proposed architecture for the cloud computing environment. The following are the main components of the proposed framework:

1) SLA Agent: The new architecture of outsourcing of services forces the business decision-makers to look for experts in IT, policy, and legislation domains. These professionals can provide services such as design IT metrics for SLA agreements, set the value for SLA parameters and examine the policy and legislations for partners. In cloud computing, SLA agents are very important as intermediary agents between consumers and cloud providers. In the proposed model, I use an SLA agent to perform the following main tasks:

i. Group cloud consumers in different classes based on business needs
ii. Design SLA metrics based on the consumers' needs
iii. Negotiate with cloud providers
iv. Select cloud providers based on non-functional requirements. The discovery and selection processes to get the cloud services based on the functional requirements are made by the consumers in the early stage of communication with cloud providers
v. Monitor business activities for consumers
vi. Monitor SLA parameters

2) Cloud Consumer: As mentioned in Chapter 3, the cloud consumer is the entity requesting the external execution of one or more services. The cloud consumer is required to pay the bill for the completed execution of services based on a well-defined pricing model. The design and discussion of pricing models for cloud computing are presented in Chapter 5 of this thesis. The SLA agent has the authority to choose the optimal price model for services. The consumer model consists of two main parts:

i. Trust management: This agent manages the trust relationships between cloud providers and also the other users of cloud services. Three sources of information are used in the trust management model: (1) the local experiences with cloud providers and users; (2) the opinions of external cloud services; and (3) the reports which are provided by the SLA agent. To obtain a reliable result from the trust management agent, I will use credibility metrics associated with these three sources of information. Cloud consumers are able to assign various weights ($0 \leq$ summation of all weights ≤ 1). The output of the trust management system will be used to rank the list of cloud providers obtained from the cloud services directory. Then, the ranked list will be sent to SLA agent to select the final cloud provider based on non-functional requirements. More details about the proposed solution for trust management in cloud computing are presented in Chapter 8 of this thesis.

ii. Business activities management: The key point of my model that distinguishes the solution from others which are designed for online services is its indicator of business activities. I propose to use this indicator for the main SLA parameters to determine who is responsible for the violation of the revenue or profit parameters. More details about these parameters are presented in Chapter 5.

3) Cloud Services Directory: Consumers of cloud services do not know about the existing cloud providers if there is no agent or registry to advertise the descriptions of their services. In my proposed architecture, I use a common directory in order to help cloud consumers to find specific services. I assume that the directory will store at least the IDs of cloud providers and the functional advertisements of their services. Here, I do not consider the processes of discovery and service selection in detail. So, the research scope in this part of my thesis is limited to the design of SLA agreements and trust management only and does not take into account cloud service discovery, selection and composition.

4) Cloud Providers: As mentioned in Chapter 3, cloud providers are the entities who own the cloud infrastructure and provide cloud services for consumers.

4.4 OVERVIEW OF THE SOLUTION FOR PERFORMANCE MEASUREMENTS FOR CLOUD SERVICES

A new paradigm of technology which enables users to access and configure a huge amount of computing resources is known as cloud computing architecture. It is very easy to use and can be implemented using a number of resources, thereby making it very popular. Hence, major providers such as Salesforce and Amazon are already offering this new technology. Cloud computing is very attractive for a wide variety of users including researchers and government organizations because the maintenance of the infrastructure is the decision of the cloud providers. A lot of attention is thus paid to the model by the cloud provider. However, because some virtual machines (VMs) may attain a magnitude of demand greater than that of the other VMs, there is the possibility of a decrease in performance [1]. This fact can indeed influence the performance of other applications considerably. This can be illustrated with an example. It can be inferred that the performance of EC2 differs to a great extent.

There are several reasons for these inconsistencies in the performances of VMs. The reasons include the conflict for the VMs like the response time which is also one of the prime reasons for this variability in the performance. This irregularity in the performance is in fact one of the major issues of cloud computing which many users face and is also considered as one of the major barriers to the success of cloud computing [1]. For example, for any applicable service, users expect similar performance and that too at any time. This is not concerned with the existing workload present on the cloud for particular application or service. Also, it is quite important for decision makers because there might be an underlying pattern or variability in performance that needs to be factored in during decision making. Another good example is that clouds depend on SLAs like a grid computing that has to be constructed within a stipulated time. Hence, enterprises also expect cloud providers to guarantee a quality of service. Thus, it is critical that cloud providers ensure that the service level agreement is based on performance features which include storage size as well as level of security. However, the general tendency of cloud providers is to create their SLAs on the basis of the trust level of the services they are offering [2-4].

Thus, currently it is obvious that there are fewer users because they have to deal with the irregularity of the performance of the cloud services. There is a need for a mechanism to enable users to deal with the same so that they are better able to understand the difference in the performance of the cloud.

This part of the solution proposed in this thesis will focus on the issues mentioned above and assess the performance level of the instances of Amazon EC2,in depth, as it is one of the best cloud networks known to date. The major contributions that I make in this chapter are as follows. First, I conducted the experiments in different scenarios, on a single VM. This can provide an estimation of the differences in performance of a single VM. Subsequently, I conducted various types of VM experiments in multiple cases of operation. This provides an estimation of the variability in performance in the case of multiple VMs. To test the performance in different locations, I conducted different experiments using two locations that provide an estimation of different levels of performance of the cloud provider's infrastructure which was in various locations. The second contribution is my analysis of the results that have been obtained using the statistics concepts to quantify the differences in cloud service performance. In this case, I increase the number of VMs and measure the difference in performance level in order to compare it with low level of performance in fewer VMs. Also, I compare the performance in these scenarios with real applications of cloud computing. The third contribution is an analysis of results and a defined methodology to divide the performance level into two segments. I use various factors but the focus will be on VMs provided by Amazon. Some recommendations have also been provided to the users so as to reduce the variability in the performance.

It is expected that the study conducted will have a great impact on the practice for three main reasons. They are:

i. The decision makers obtain a good understanding of the results which have been produced by conducting experiments using this well-known cloud provider.

ii. A deeper understanding of the service level agreements is obtained from Amazon EC2 indicating what users can be offered.

iii. Suggestions have also been provided about what can be done to reduce the variance in the performance.

The research tasks for the methodology of the performance measurements solution have been defined as follows. I start with a survey which presents the existing methodologies that have been proposed for performance evaluation of cloud services. A detailed overview of the Amazon EC2 cloud has been provided in order to present basic knowledge about the architecture of Amazon EC2. Then, I discuss the interesting findings of the performance of

Amazon EC2, and the different benchmarks used in the experiments are presented. The results are then discussed in detail followed by the analysis of the variability in the performance as the resources increased and also with real applications. Further in this part of thesis, users of cloud computing have been provided with some advice so that the results obtained from experiments are meaningful and significant on the Amazon platform. The detailed solution is presented in Chapter 7.

4.5 OVERVIEW OF THE SOLUTION FOR TRUST AND REPUTATION FOR CLOUD COMPUTING

Cloud computing involves a trade-off between costs, benefits, security and privacy. Trust is a key to acceptance of Software as a Service. Unless the customers and organizations can trust that cloud providers will protect the security of sensitive information, potential customers might refrain from using cloud services. Secure handling of data, accountability and privacy safeguard promote trust among users and service providers and also provide the basis for acceptance of cloud computing services.

In conventional models of internet security, a security boundary is deployed to build a trust perimeter to control the use of computing resources. In such a model, the customer or organization also has the control of storing and processing the data as they please. However, this is not possible in cloud computing and the security boundary is compromised because the data is processed on machines that are owned and controlled by someone else. The contractors or sub-contractor may also process the classified data outside the trusted vendors thereby increasing the risk of illegal uses, resale or leaking of sensitive data.

Thus, cloud computing has certain serious issues and its use is risky unless several important measures are taken to avoid these risks.

My methodologies in this thesis are proposed to reduce such risks in the cloud community. I use the concepts of economics, service level agreements (SLAs), evaluation of performance, and trust management to provide different contributions to cloud computing. These contributions are discussed and evaluated in order to assist cloud providers and cloud customers to obtain more advantages in the cloud community. More details about my proposed solution for trust and reputation for cloud services are presented in Chapter 8.

Trust is a very important factor in the open distributed systems. It is the main concern of all interactions between service providers and consumers in such a changing environment. Trust evaluation is not a clear and easy process for different users because it has vague and different subjective values. Hence, there is the need for a description methodology that presents the values of trust in a clear way. Fuzzy logic is an effective technique to solve this problem. The fuzzy logic provides a basis for modelling human perceptions using a better description method. Therefore, my solution for trust in cloud computing will use the fuzzy logic approach to evaluate the trustworthiness of its service providers. Figure 4.2 shows the main steps in the trust evaluation process for cloud computing using the concepts of fuzzy logic approach.

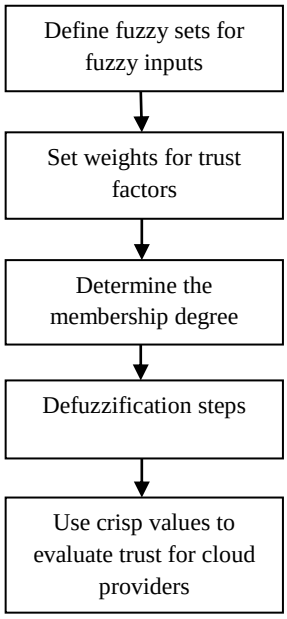


Figure 4.2: Trust evaluation using fuzzy logic approach

In the proposed framework of trust for cloud services, I develop an approach that takes into account the key aspects of the trust relationship between cloud providers and users. Further, each of the trust dimensions will be represented within a fuzzy framework, and measures along each dimension are developed. In addition, an overall figure of trust value will be developed for the cloud providers. The detailed solution is presented in Chapter 8.

4.6 CONCLUSION

In this chapter, I provide an overview of the proposed solution and the methodology that I will adopt in order to solve the problem presented in Chapter 3. In Chapters 5-8, I give the details of the four solutions that have been proposed. I discuss first how the dynamic pricing solution can be used to solve the correlation problem with dynamic allocation of cloud resources. Then, I show the proposed service level agreements and how these can be used by the cloud community. Also, I provide an overview of the performance measurement methodology that will produce results enabling the users of cloud computing to monitor cloud service objectives. Finally, I consider the use of trust and reputation concepts to evaluate the trustworthiness of cloud computing agents using a fuzzy logic approach.

The next four chapters present my proposed methodologies in more detail along with the evaluation results.

REFERENCES

[1] J. Joutsensalo, T. Hämäläinen, K. Luostarinen, and J. Siltanen, *"Adaptive scheduling method for maximizing revenue in flat pricing scenario",* AEU-International Journal of Electronics and Communications, vol. 60, pp. 159-167, 2006.

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, and I. Stoica, *"Above the clouds: A berkeley view of cloud computing",* EECS Department, University of California, Berkeley, Tech. Rep, UCB/EECS-2009-28, 2009.

[3] G. Wang and T. S. E. Ng, *"The impact of virtualization on network performance of amazon ec2 data center",* Proceedings of the 29th conference on Information communications, pp.1163-1171, San Diego, California, USA, 2010.

[4] D. J. Abadi, *"Data management in the cloud: Limitations and opportunities",* IEEE Bulletin on Data Engineering, pp. 3-12, 2009.

[5] I. Foster, Y. Zhao, I. Raicu, and S. Lu, *"Cloud computing and grid computing 360-degree compared",* Grid Computing Environments Workshop, pp. 1–10, 2008.

CHAPTER 5 – A DYNAMIC PRICING MODEL FOR CLOUD SERVICES

5.1 INTRODUCTION

There are different domains where cloud services are available including government departments, small businesses, social users, and the education sectors. All of these sectors are connected through the internet and they can rent resources offered by cloud services providers. This cloud-based resource has been discussed in detail in the previous chapters. All these sectors have different objectives for connecting with the cloud providers which in turn will affect the pricing of the corresponding cloud service and the revenue of cloud providers. Hence, there is the need to set up a dynamic pricing model for the cloud services or resources that each sector will require. Since this type of pricing method will give the consumers a better way of paying for the consumed services, it is relevant to discuss in detail the benefits of, and expectations from, such a pricing model. Every form of business or service that is offered always involves a pricing scheme. The developing and operating model of cloud services makes its pricing scheme a little complex. The pricing mechanism is what decides how service customers are charged [1]. There are several different factors on which the pricing of a service can be based. Some customers may be charged based on the time when the request was filed; this is known as the "peak/off-peak" pricing. There are also "fixed" price rates, as well as "supply/demand" where the pricing will rely on the resources that are available. Importantly, an effective pricing scheme is used to determine the supply and the demand by managing the computing resources that are found within the data centres. By determining the pricing factors, this helps with the allocation of the available resources along with the pricing of the corresponding resources.

A dynamic pricing mechanism in cloud services is responsible for maintaining the actual usage of resources [1]. The requests are allocated to the proper resources and when this is done, the cost of doing so is charged to the customer [2]. A dynamic pricing method also takes into account the historical usage information. This stored information can be utilized by scalable admission control. Information like this is helpful in improving the resource allocation decisions. This chapter will discuss different schemes that can be applied for

pricing services that are being offered by online providers. Then, I present a dynamic pricing scheme for cloud services. Finally, I analyse the results and evaluate the proposed scheme.

5.2 DYNAMIC PRICING SCHEMES

There are three pricing schemes which can be used for pricing of cloud-based resources. These schemes are:

A. Fixed pricing

At first, users use a small amount of internet resources. To increase the market, a unified price package is used. With this, for a certain time period, the users have the same access speed and price. This is a simple system in which the charges are known to the users and this system is quite common in the internet access services market [3].

However, this scheme has several disadvantages. The users have no idea of how to adapt their usage models, and they over-use the online resources. Then service providers apply the same pricing method to all users regardless of their individual usage. This does not increase the overall development of the distributed technology, and the performance of the system is degraded. As the technology has become further developed (in areas such as online applications and the complex e-commerce environment) this model no longer applies.

B. Resource-based pricing

This model charges the users independently. The charges are calculated on the basis of the amount of transferring and receiving data on network. The service provider uses statistical sampling methods to assess the usage. The charges are applied according to the demands of the supply. The interaction between service customers and service providers is also developed, unlike the fixed pricing model. It enhances the consuming approach efficiency of service resources and controls the bottleneck of the usage. But with the development of services requiring high-bandwidth, the overall user's allocation of resources has increased together with an increase in charges. In terms of content sharing in some charging methodologies, this has become a problem for online-based service providers to charge according to the consumption of resources.

Authors in [4] suggested a complicated pricing model. In this approach, customers prefer traditional services at a fixed rate but higher demands on resources are charged on the basis of the usage. The method of filtering and counting the packets of data for large number of

users increase the complexity of this approach. The analysis of the simulation results shows that the complicated price modelling approach can lead to an improvement in network resource performance while increasing the service provider's revenue [4].

C. Blocking pricing

With the problems of high resource applications, privacy issues and extra costing problems, the network blocking has increased. This has created a bad effect called 'cost-based' allocation for the service users. So a new model, congestion scheme of pricing, has been developed[5], which groups the price so that it reflects the service resource usage and the service cost. It encourages the users to define rules for the service demand. In this way, the blocking problems are addressed. The price fluctuates dynamically according to congestion.

But cost-based allocation is difficult to measure as each user's value of service resources is different. A dynamic-based pricing approach was proposed to measure the cost-based allocation. Several other methods have also been introduced, and the main goal is to determine a price-aware service resource system that shifts the amount of load from a time of high load to a time when service resources are stable. In this way, the blocking of traffic can be under control and produce some level of reliability [5]. In the next section, I present my proposed approach for pricing of cloud-based resources.

5.3 DYNAMIC PRICING MODEL FOR CLOUD SERVICES

Distributed computing systems such as grid computing, peer-to-peer, and the new cloud technology create a new way of using and sharing computing resources over the Internet. In cloud computing, users reserve virtualized resources in different capacities without any knowledge of the complicated allocation method of the underlying infrastructure of cloud data centres. For Infrastructure as a Service (IaaS), customers pay the service provider using a static scheme. Dynamic demand of cloud computing is a key feature of the new paradigm of distributed services. Using a fixed price for cloud services does not give more benefits for cloud providers in terms of optimizing resource allocation approach for cloud infrastructure. Also, a fixed price for cloud services may force customers to use services of other competitors who provide services at a lower price and higher level of quality.

My proposed model for dynamic pricing for cloud computing contains three main entities: cloud customer, cloud provider, and cloud market agent. Cloud customers propose their prices for desired cloud-based resources. Also, the cloud provider presents the price of its provisioned services to the cloud market agent. The cloud market agent is responsible for matching the service price proposed by the cloud provider with that of the customers that announce the same level of service price. In the second step, the cloud market agent matches the service level objectives (from the cloud customers) with the service performance of the cloud providers. These steps lead to the reliable monitoring system to monitor the performance of cloud resources. Also, the proposed SLA agreements definition presented in Chapter 6 can be used in this process in order to link the service price with the actual level of quality of cloud services. If there are no resources available to meet the customer request, the request will be placed in an admission queue until the resources of the cloud provider that can fulfil the customer requirements are available.

The pricing model components are shown in Figure 5.1.

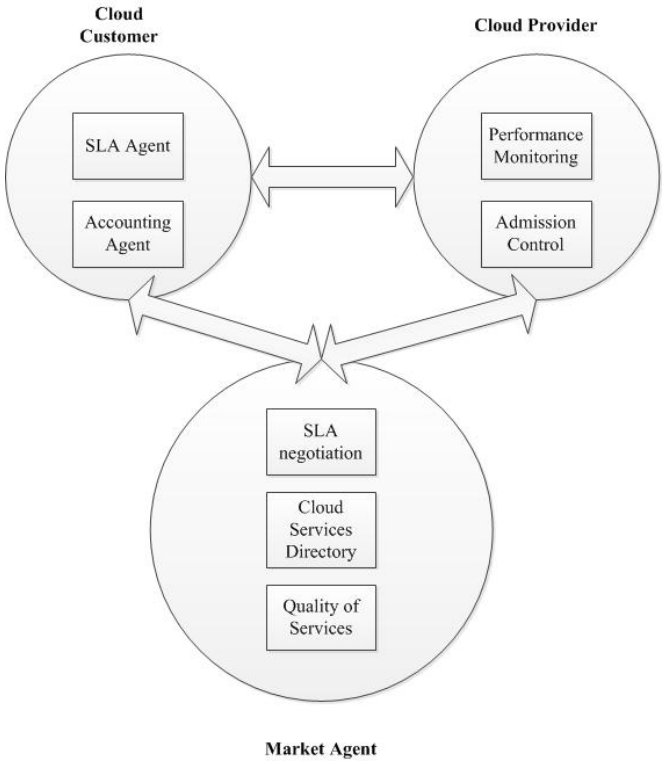


Figure 5.1: Pricing model components

In cloud computing, the price of a service is determined by the scalability of resources, the market demand of the corresponding resource, and the willingness of cloud customers. In the cloud computing market, the priority of cloud providers is to maximize the service price. However, cloud users want to minimize the price and receive a high quality of service. So, any pricing model for cloud computing should take these issues into account and provide a solution that satisfies both parties in the cloud services market.

5.3.1 Pricing Scheme from The Cloud Customer's Perspective

For the dynamic pricing model proposed in this thesis, I define the following approach to allow cloud customers to retain the history of service prices for a specified period of time. This can help customers to compare the current prices of services with the new prices proposed by service providers.

$$CL_{pr}(t) = \mathrm{Pr}_{-1} + pr_{Dif} \qquad (1)$$

Where, $CL_{pr}(t)$ is the existing price and Pr_{-1} is the previous price of the cloud service proposed by the service user. pr_{Dif} is the amount that can be used to increase or decrease the service price. The value of pr_{Dif} depends on the market demand and the quality of the cloud service. pr_{Dif} can be calculated by using a weighting variable to determine the degree of variance in the new price of the service. In my proposed dynamic pricing model, I use the following equation to calculate pr_{Dif} from the perspective of the cloud customer.

$$pr_{Dif} = \lambda(x(t) - thv)\,\mathrm{Pr}_{-1} \qquad (2)$$

Where, λ is the weight to determine the degree of changing the service price; x(t) is the function that is used to evaluate the quality of the cloud service from the cloud customer's perspective; thv is a threshold value for the quality of service that can be used to fix the level at which the customer is not able to accept the price with the quality level under this value. The algorithmic formulation of my dynamic cloud service/resources pricing approach is presented in Algorithm 5.1

Algorithm 5.1: Dynamic pricing algorithm for cloud customer

Cloud customer (Dynamic pricing algorithm)

Input: $\Pr_{-1}$, pr_{Dif}, λ, n

Output: Current proposed price CL_{pr}

Begin

Set initial value for $\Pr_{-1}$

For i = 1 **to** n **do**

Calculate $pr_{Dif} = \lambda(x(t) - thv)\Pr_{-1}$

Calculate $CL_{pr}(t) = \Pr_{-1} + pr_{Dif}$

end

The optimal time of renting cloud services

Users of cloud computing can use my proposed approach to determine the optimal time to rent services provided by different cloud platforms. In the context of selecting the time when a service will be used, I discuss in general when cloud customers can receive a high level of quality and pay a good price for short-term use of cloud services. The QoS parameter is very important in the context of distributed services. So, users of cloud services are looking for QoS at the period of some selected hours. A detailed investigation of this problem is not within the scope of this thesis. Figure 5.2 shows the level of quality of cloud services at different hours of the day.

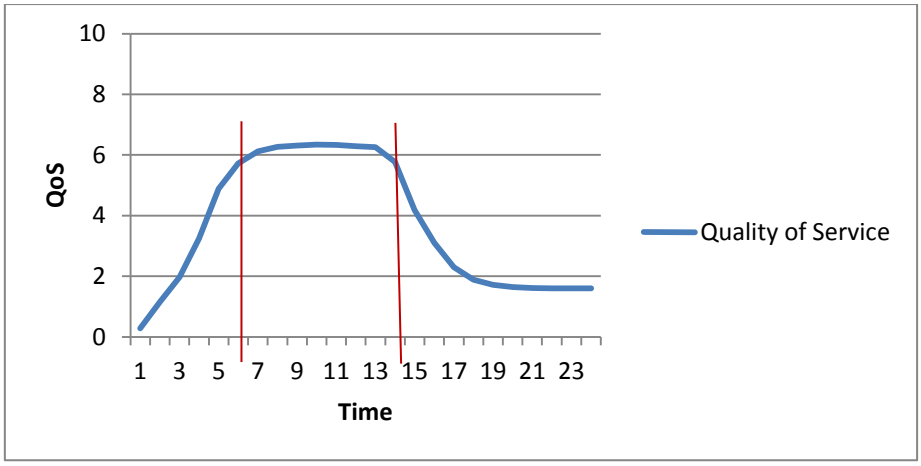


Figure 5.2: QoS of cloud services at different hours of the day

Customers of cloud computing have different objectives when using cloud resources. Sometimes customers can delay their use of services by a different number of hours or days. In this case, my proposed dynamic pricing model can be used to propose a price level for the cloud services based on the quality during a given period of time. For example, as shown in Figure 5.2, the peak period interval can be avoided by using a service during the off-peak period. In this case, users can receive a high level of quality at a relatively lower price.

5.3.2 Pricing Scheme from The Cloud Provider's Perspective

The cloud service provider publishes the price of service for the cloud market agent in order to attract cloud customers at different periods of time. The cloud market agent uses the advertised price in the negotiation and matching processes. In my proposed model of dynamic pricing for cloud computing, the cloud service provider uses the following equation to announce prices based on the time period of the market:

$$PL_{pr}(t) = \mathrm{Pr}_{-1} + pr_{Dif} \qquad (3)$$

Where $PL_{pr}(t)$ is the existing price and Pr_{-1} is the previous price of the cloud service proposed by the service provider. pr_{Dif} is the amount that can be used to increase or decrease the service price. The value of pr_{Dif} depends on the market demand and the function of resource

load for the cloud data . pr_{Dif} can be calculated by using the weighting variable to define the degree of variance in the new price of service. In my proposed dynamic pricing model, I use the following equation to calculate pr_{Dif} from the perspective of the cloud provider:

$$pr_{Dif} = \sigma(x(t) - thv)\,\mathrm{Pr}_{-1} \qquad (4)$$

Where σ is the weight to determine the degree of changing the service price and x(t) is the function that is used to determine the load degree of cloud computing resources. *thv* is a threshold value for the cloud resources load that can be used to fix the level so that customers are not able to obtain the current price under this value. The algorithmic formulation of my proposed approach for pricing of cloud resources from the perspective of the cloud provider is shown below in Algorithm 5.2.

Algorithm 5.2: Dynamic pricing algorithm for cloud service provider

Cloud service provider (Dynamic pricing algorithm)

Input: Pr_{-1}, pr_{Dif}, σ, n

Output: Current proposed price PL_{pr}

Begin

Set initial value for Pr_{-1}

For i = 1 **to** n **do**

Calculate $pr_{Dif} = \sigma(x(t) - thv)\,\mathrm{Pr}_{-1}$

Calculate $PL_{pr}(t) = \mathrm{Pr}_{-1} + pr_{Dif}$

end

5.4 SIMULATION AND DATA ANALYSIS

In order to simulate the proposed dynamic scheme model, I conducted two multi-step experiments. The first experiment investigated the method of allocation of cloud resources by using the proposed dynamic scheme and the static pricing scheme. The second experiment is to study the revenue produced by my method of dynamic pricing over a given period of time and compare the revenue in this case with the revenue produced by that of the static pricing method.

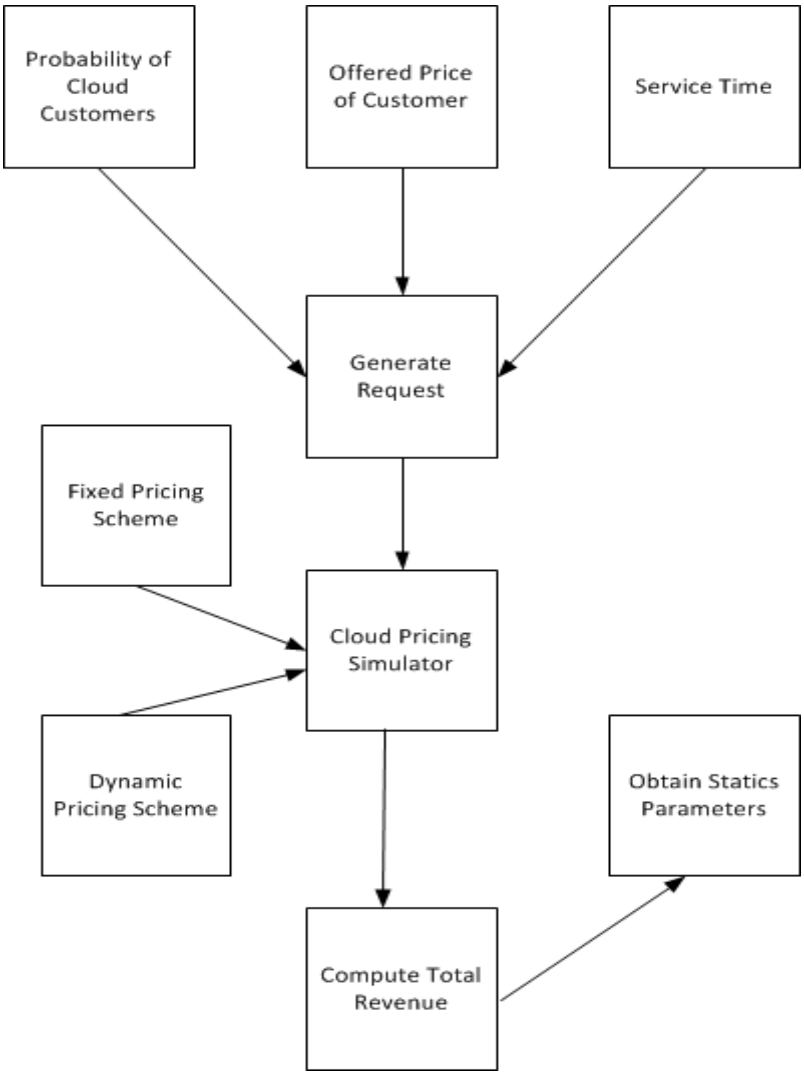


Figure 5.3: Simulator

Figure 5.3 shows the steps of the simulation experiments that are used to validate the proposed scheme for dynamic pricing of cloud services. The inputs of the simulation model

are: service time slot, offered price of given customer, and the probability of cloud customers of the cloud market. The first step of the simulation is to generate cloud customer requests using model inputs. I use two sets of pricing schemes. The first one is the fixed pricing scheme in which there is no correlation between the proposed price of cloud services and the way that cloud resources are allocated. The second scheme is the dynamic pricing scheme which I propose in this thesis. In each set of pricing schemes, the simulation model computes both a proposed price for a given cloud service and the total revenue, and obtains statistical parameters such as performance metrics. In Sections 5.4.1 and 5.4.2, I use these parameters to analyse the differences between both schemes of pricing for cloud servicers.

5.4.1 Optimizing Resource Allocation of Cloud Services

The rate of customer numbers arriving with the static pricing method is presented in Figure 5.4.

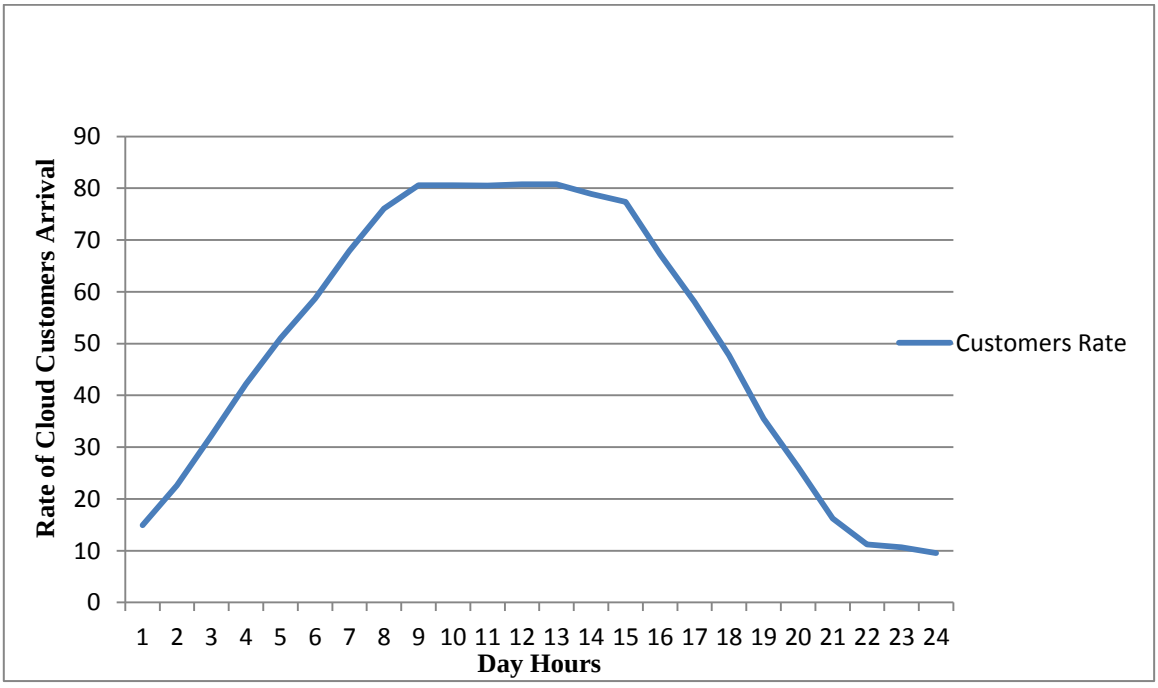


Figure 5.4: Optimizing resource allocation of cloud services

I observe from figure 5.4 that the resource load of the data is not uniform. The resources load rate early in the day is low because customers do not use their full capacity of cloud resources at this period of time. I consider the CPU capacity to be the bottleneck of cloud computing resources. To solve the problem of having cloud resources bottleneck during on-peak periods, the cloud provider could charge different prices for different times of the day. The correlation between cloud service prices and the demand of customers is more complicated in terms of the competitive market, virtualization, and performance of cloud data centres. All these aspects need to be considered in any proposed mechanism for cloud service pricing. The cloud provider can use differentiation in service pricing methods to control the admission of customers' requests during on-peak and off-peak periods. The high price of services forces many customers to postpone their request to the off-peak period. In this case, cloud resources can be optimally managed and allocated and at the same time, the revenue of the cloud provider can be maximized with most possible number of requests. Also, decreasing the price of services during off-peak periods could encourage more customers to use cloud resources during such periods. A dynamic pricing scheme provides a balancing policy to maintain the level of availability in homogeneous cases during different periods of time as presented in Figure 5.5.

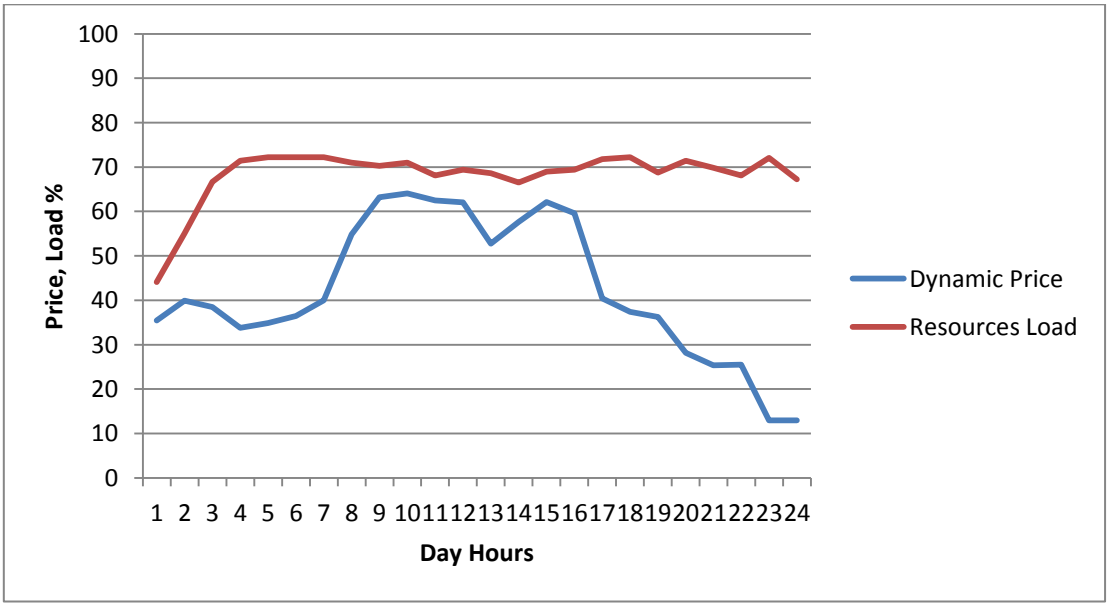


Figure 5.5: Impact of cloud service pricing scheme on resource load

5.4.2 Total Revenue of Cloud Service Provider

I simulate the market of cloud services and collect the total revenue for the cloud provider using simulations with both fixed price and dynamic pricing schemes. These simulations were conducted over 24 hours on different days during several weeks. As shown in Figure 5.6, the total revenue using the dynamic pricing scheme is greater than the revenue using the fixed pricing method. This is because my proposed method provides an efficient methodology for optimizing resource allocation of cloud data with no effect on the quality of services provided for different classes of users. The use of a flexible and dynamic method for cloud services results in more customers being willing to rent cloud services at different periods of time. On the other hand, the fixed method of pricing services can discourage customers from using the cloud infrastructure at bottleneck load system. From the perspective of the cloud provider, this could place business continuity at risk.

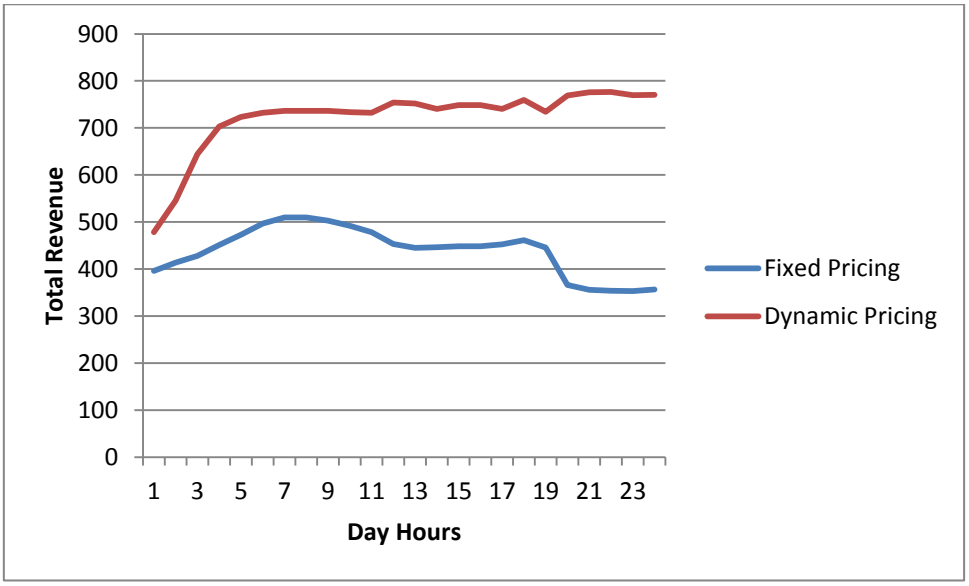


Figure 5.6: Total revenue for cloud service provider with different schemes of pricing

As the result of fixing the price of services, the quality of service is not acceptable to customers, especially during the on-peak period. In order to evaluate the satisfaction level of the proposed method using the fixed pricing approach, I conduct an Arrival-Request, Total Revenue experiment. The output of this simulation is presented in Figure 5.7. I notice that the rate of request arrivals using the dynamic pricing method produces significant improvements in the resource allocation of cloud services.

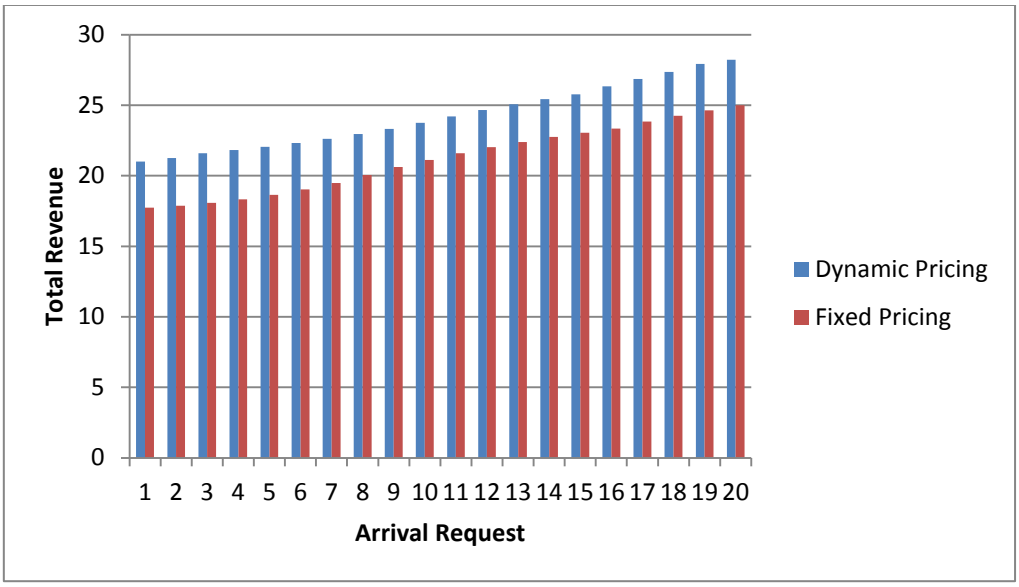


Figure 5.7: The impact of total requests on the total revenue for cloud service provider

5.5 CONCLUSION

In this chapter, a dynamic pricing scheme is introduced for cloud service customers and cloud service providers. In the proposed scheme, a cloud market agent is used to provide the matching process and negotiation about service level objectives for cloud users. In this scheme, pricing functions are proposed to control the cost level from the perspective of cloud customers and to control the resource allocation and maximize the revenue from the perspective of cloud service providers. I discussed the problem of how to announce the service price when the demand for services and the resources are not stable in the cloud computing market. I use weighting parameters λ and σ to control the increasing and decreasing amount of pricing when the cloud market is in high and low phases. The conducted simulations show that the proposed dynamic pricing scheme for cloud computing provides scientific results of improving the revenue for cloud service providers and help cloud customers to rent high quality cloud services.

In the next chapter, I use SLA agreements to correlate the proposed method in this chapter with the concept of using an SLA definition for different cloud service providers.

REFERENCES

[1] P. Kannan and P. K. Kopalle, *"Dynamic pricing on the Internet: Importance and implications for consumer behavior"*, International Journal of Electronic Commerce, vol. 5, pp. 63-83, 2001.

[2] K. L. Haws and W. O. Bearden, *"Dynamic pricing and consumer fairness perceptions"*, Journal of Consumer Research, vol. 33, pp. 304-311, 2006.

[3] J. Joutsensalo, T. Hämäläinen, K. Luostarinen, and J. Siltanen, *"Adaptive scheduling method for maximizing revenue in flat pricing scenario*", AEU-International Journal of Electronics and Communications, vol. 60, pp. 159-167, 2006.

[4] M. L. Honig and K. Steiglitz, *"Usage-based pricing of packet data generated by a heterogeneous user population"*, INFOCOM '95, Fourteenth Annual Joint Conference of The IEEE Computer and Communications Societies, Bringing Information to People. Proceedings. IEEE,vol. 2,pp. 867-874, 1995.

[5] Y. Hayel and B. Tuffin, *"An optimal congestion and cost-sharing pricing scheme for multiclass services"*, Mathematical Methods of Operations Research, vol. 64, pp. 445-465, 2006.

CHAPTER 6 – SERVICE LEVEL AGREEMENT FRAMEWORK FOR CLOUD SERVICES

6.1 INTRODUCTION

As the era of cloud computing has changed the way that online services are provided, service level agreements are considered to be an important key factor in providing cloud services, with an acceptable level of reliability and security. This chapter provides a framework to define customized structures for service level agreements to be used between service users and service providers in the cloud services market.

6.2 DESIGN CRITERIA FOR SLA FRAMEWORK FOR CLOUD SERVICES

This section presents the design criteria of the proposed framework of an SLA for cloud services. After reviewing the literature related to SLAs in different domains, I identify the following design criteria to be considered in the design phase of the SLA framework. Also, these criteria are used as performance criteria in the validation phase of my thesis to evaluate the proposed framework in the context of cloud computing.

1. **Scalability**

   Cloud computing provides a framework for accessing various types of virtual machines from a scalable pool of cloud resources. These resources include hardware, network resources, development tools, and databases. Cloud resources need to be developed in a dynamic and scalable way taking into account the variable workload performed by cloud users. Scalability is “the ability of a system, network, or process, to handle growing amounts of work in a graceful manner or its ability to be enlarged to accommodate that growth” [1]. The scalability feature is an essential consideration in any proposed solution for cloud services. I propose an SLA framework for cloud services that can be dynamically scaled up and scaled down to provide optimized allocation of cloud resources to the cloud users. Also, this feature will reduce the cost

of services from the perspective of cloud customers when they no longer need to use a virtual machine.

2. **Usability**

The usability of any system is essential to increase the satisfaction of users and reduce supporting costs. If SLA management systems are developed and implemented without a sufficient level of usability, this could extend negotiation time and cause problems in mapping service objectives to SLA parameters. Especially in cloud computing, the usability factor is one of the most important factors to be considered when developers want to create cloud-based solutions. The customers of cloud services may not necessarily be aware of the complicated technology of the cloud infrastructure since they use cloud services like outsourcing solutions and pay for what are they using. In my proposed SLA framework for cloud services, I consider this issue in order to produce SLA templates for different types of cloud services.

3. **Mapping SLA parameters**

Cloud services are provided via a large variety of resources. These resources can be monitored by measuring the performance parameters to be used in SLA templates. Due to the large number of performance parameters of cloud resources, cloud users face the problem of mapping SLA parameters to service objectives [1]. In some cases, SLA developers define the SLA structure inappropriately when matching service objectives with the correct parameters of performance. This problem can affect the continuity of the business model of cloud customers and create problems when assigning responsibility for the violation of terms. The proposed framework in this thesis aims to establish a reliable method of mapping objectives of cloud services to the appropriate performance parameters in order to maximize the level of satisfaction of users and minimize the risks associated with the use of cloud services.

6.3 SLA LIFE CYCLE

In this section, I propose a model for an SLA life cycle for my framework. Ron et al. [1]'s model of the SLA life cycle comprises the following phases:

- Creation phase
- Operation phase
- Removal phase

For my proposed framework, I map the three phases of the life cycle model proposed by Ron et al. and add three more phases. The life cycle of my SLA framework for cloud services includes the following phases:

- Discover cloud service
- Define SLA
- Establish SLA
- Monitor SLA metrics
- Remove SLA
- Execute penalties

Figure 6.1 shows the life cycle of my SLA framework for cloud services.

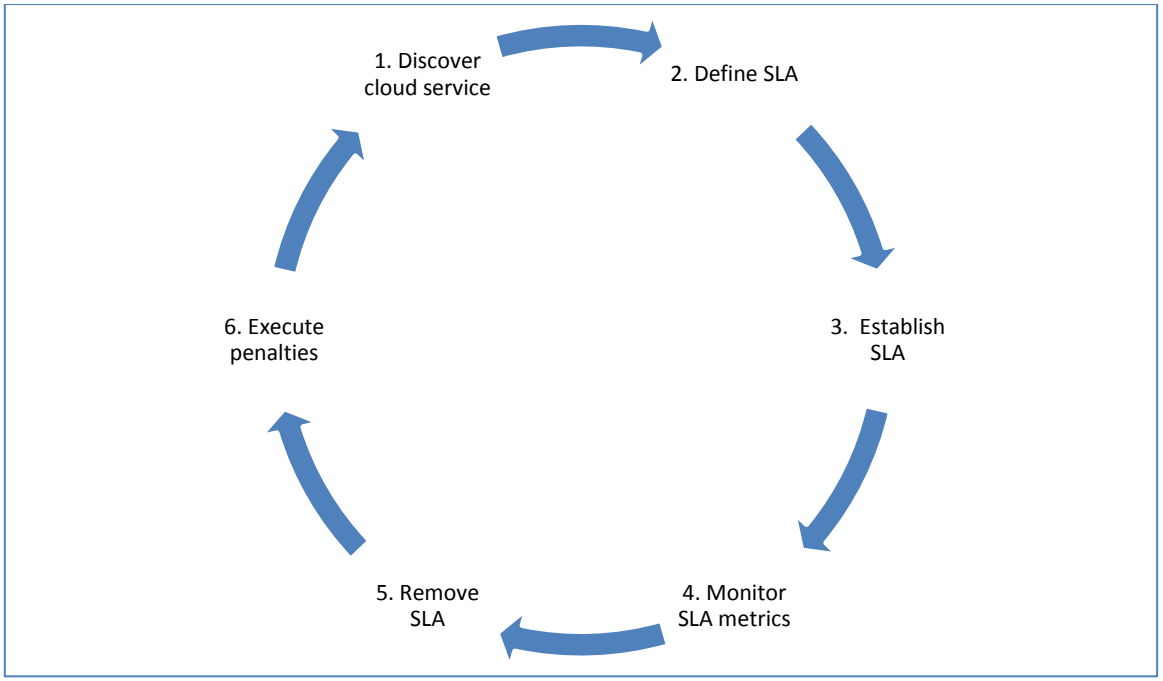


Figure 6.1: SLA life cycle

The first step of my proposed SLA life cycle framework is discovering the cloud service. In this step, the service consumer provides service requirements to obtain a list of service providers who can meet the terms of the required service. In the second step, the SLA is defined with service objectives and QoS criteria. The third step is to establish the SLA agreement and the service users are able to start using their allocated resources. In the fourth step, the SLA parameters are monitored so that they can be compared with the agreed values of the SLA content. In step five, the SLA can be terminated in any case of violation or because the service time has ended. The last step is the execution of penalty phase where the penalty corresponding to the SLA violation will be executed.

6.4 SLA FRAMEWORK FOR CLOUD SERVICES

In Chapter 3, I discussed the problem of defining service level agreements for cloud services. I stated that cloud customers should have a framework enabling them to create and deploy SLAs in a flexible and reliable way when they propose to use cloud technology. This section of my thesis describes my framework for designing an SLA in the cloud environment and defines the basic algorithm to use in a more trusted environment. Figure 6.2 describes the proposed framework of an SLA for cloud services.

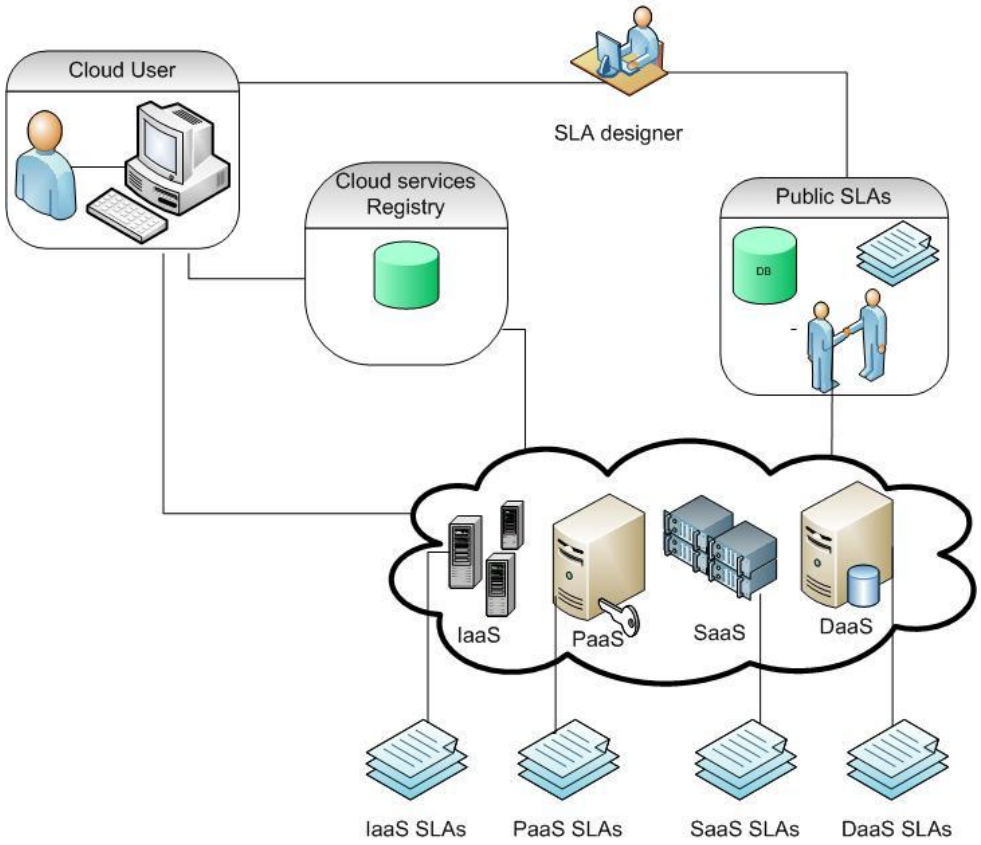


Figure 6.2: SLA Framework for cloud services

6.4.1 SLA Framework Components

Cloud service provider: Service providers may provide one of the following services:

1. Infrastructure as a Service (IaaS)
2. Platform as a Service (PaaS)
3. Software as a Service (SaaS)
4. Database as a Service (DaaS)

Cloud user: A cloud user is the user who can use cloud services in different models. Users can be individuals, small businesses, or large organizations. Cloud users may use one of the following services: IaaS, PaaS, SaaS, or DaaS.

Public SLA Pool: The SLAs pool contains all templates of SLAs which have been devised by designers who have knowledge about cloud business models and the technology of cloud services. Experts who establish the definitions of SLAs for cloud services also take into consideration the service objectives and requirements to specify the most relevant parameters of a SLA in order to provide better customer satisfaction.

SLA designer: The SLA designer is the expert who has knowledge about cloud business models and cloud services technology. SLA experts in most cases contact cloud customers and providers to obtain information about service requirements and existing results of performance parameters.

Cloud services registry: This repository contains information about cloud services and cloud providers. Descriptions of quality of services and non-functional parameters are provided by cloud providers with a clear method of costing and penalty calculation for services.

6.4.2 The Processes of SLA Framework

The processes of the proposed SLA framework start with the service request being submitted by the cloud customer to the cloud services registry. Cloud customers specify the type of cloud service and quality of service in a description document. The administrator of the cloud services registry matches the service requirements to the best cloud platform for the cloud customer. The cloud services registry sends the result to the cloud customer without any SLA document. In this stage, the administrator of the cloud services registry chooses the type of cloud platform from one of the following: IaaS, PaaS, SaaS, and DaaS. Then, the cloud customer begins communication with the SLA designer to obtain the most relevant SLA (for that type of cloud service) which is already stored in the database of the SLA designer. The SLA designer may need to retrieve the SLA template from the public SLAs registry if it is not in the local database of the SLA designer. Although the SLA designer and cloud providers have already performed many interactions to create general SLA templates, cloud customers may require more specific types of SLAs if the general SLAs do not meet the service requirements. In this case, more interactions are required between the SLA designer,

cloud customer, and cloud provider to define most relevant parameters for the SLA. In the final stage of the SLA creation processes, the SLA designer sends an SLA template to the cloud customer in readiness for the establishment process between cloud customer and nominated cloud provider.

In case of violation for SLA, service provider can apply banalities and terminate the services in order to change the service provider.

After the defining stage of the SLA, the negotiation process begins between cloud customer and cloud provider. This stage may take either a short or long time depending on several factors which include the creativity of the SLA designer, customer budget, and workload of the cloud provider. After the negotiation process, the cloud customer and cloud provider sign the SLA for a specific period of time. After this process, the cloud provider can publish a copy of the agreed SLA in the public SLA registry to be available for later public use. Then, the monitoring of SLA parameters can be performed by a third party or by a local monitoring system. Performance results can be retrieved in terms of monitoring times or may be done using a live monitoring method. Finally, the cloud provider bills the cloud customer for the consumed resources. The cloud customer pays the provider, who in turn sends confirmation of payment. In the case of terms violations, a penalty must be executed between cloud customer and cloud providers using the approach specified in the SLA. Figure 6.3 presents the sequence diagram of my proposed framework for defining SLAs for cloud services.

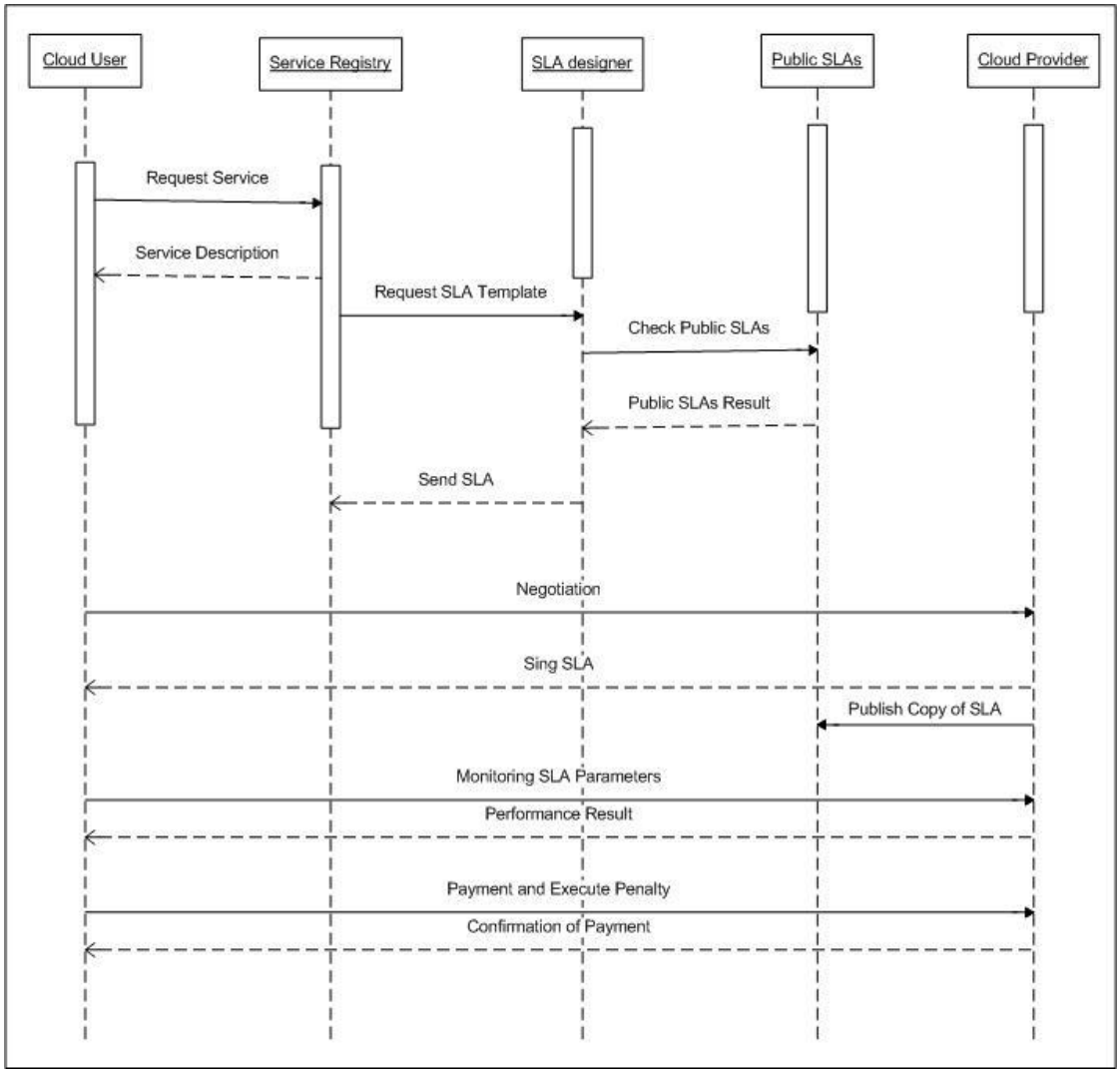


Figure 6.3: Sequence Diagram of SLA Framework

6.4.3 Correlation of Quality of Services and SLA

The correlation concept has been adopted in many domains of services. The purpose of using the correlation approach in the domain of e-services is to analyse the gap between two levels of services for a given period of time. In the context of cloud computing, I propose to use correlation to define a relationship between service level metrics and the common value of reputation of a given service provider. For example, monitoring quality of service criteria such as availability, response time, and usability can be correlated to predict the reputation of that service provider.

Cloud services that meet user requirements should be selected dynamically from existing cloud platforms based on the specified level of SLA parameters. In order to evaluate the criteria of an SLA, an efficient calculation method is required to produce the weighting result of all parameters of the SLA. In my proposed solution, I extend the methodology proposed by Taher in [2] to be used in cloud computing. I assume cloud services that meet user

requirements are denoted as CS. $CS_{i=}(CS_1, CS_2, CS_3, ..., CS_N)$. Considering J number of SLA parameters, I can use following matrix:

$$QCS = \begin{bmatrix} p_{1,1} & p_{1,2} & .. & p_{1,m} \\ p_{2,1} & p_{2,2} & .. & p_{2,m} \\ : & : & : & : \\ p_{n,1} & p_{n,2} & .. & p_{n,m} \end{bmatrix}$$

Where $P_{i,j}$ is a parameter of SLA. In order to calculate the final value of the SLA for a cloud service, the matrix QCS can be normalized to be uniform with the level of criteria. To do the normalization, two equations are used. The first one is:

$$Nor(p_{i,j}) = \frac{Max(p_{i,j}) - p_{i,j}}{Max(p_{i,j}) - Min(p_{i,j})} \quad (1)$$

The second equation is:

$$Nor(p_{i,j}) = \frac{p_{i,j} - Min(p_{i,j})}{Max(p_{i,j}) - Min(p_{i,j})} \quad (2)$$

To normalize the given parameter values, I need to define the case of benefits (denoted CB) which (CB_i= 0,1) for cloud service customers. In this case, equation 2 will be used to normalize the parameter value. If the increased value of a given parameter (say i) benefits the cloud service customer, then CB_i= 1. If the decreased value of a given parameter (say i) benefits cloud customers, CB_i = 0, then equation 1 will be used to normalize the parameter value. For example, the boot time of a virtual machine must be normalized by equation 1, and

the availability of a cloud service must be normalized using equation 2. The key idea of using the normalization method is to help cloud service providers to select which SLA has a greater effect on their resources allocation. Also, from the perspective of service customers, the normalization method will help users to select the most relevant provider from different levels of cloud providers.

The proposed methodology for correlating the SLA parameters of cloud services comprises the following steps:

1. Define the functional requirements of cloud customers
2. Select cloud service models (IaaS, PaaS, SaaS, DaaS)
3. Define SLA
4. Assign values for SLA parameters
5. Normalize p_i using the above equations
6. Calculate the final weight of SLA

To calculate the final weight of the SLA, I use the method proposed by Chang [3]. In the cloud computing environment, I use the following equation to calculate the final weight of the SLA:

$$FW_{SLA} = \sum_{c=1}^{N} p_c \times weig_c \qquad (3)$$

Where, FW_{SLA} is the final weight of the SLA from the perspective of the cloud provider. P_c is the SLA parameter, and $weig_c$ is the weight of p_c parameter assigned by the service provider.

6.5 SLA DEFINITION FOR CLOUD SERVIES

In my proposed framework, the SLA parameters are specified by metrics. These metrics define how cloud service parameters can be measured and specify the values of measurable parameters. In the cloud computing architecture, there are four types of services which providers can offer to consumers. These services are: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), and Database as a Service

(DaaS). The proposed SLA metrics for cloud computing consider these four types of these services. In this section, I define the most important parameters of SLA that consumers can use to create a reliable model of negotiation with the service provider.

6.5.1 SLA Metrics for IaaS

Companies like Amazon.com provide Infrastructure as a Service. Most of the consumers are confused as to which important parameter should be defined in the hardware part of the SLA. I list the most important parameters for consumers who are interested in using cloud as an infrastructure service. Figure 6.4 shows the IaaS SLA metrics.

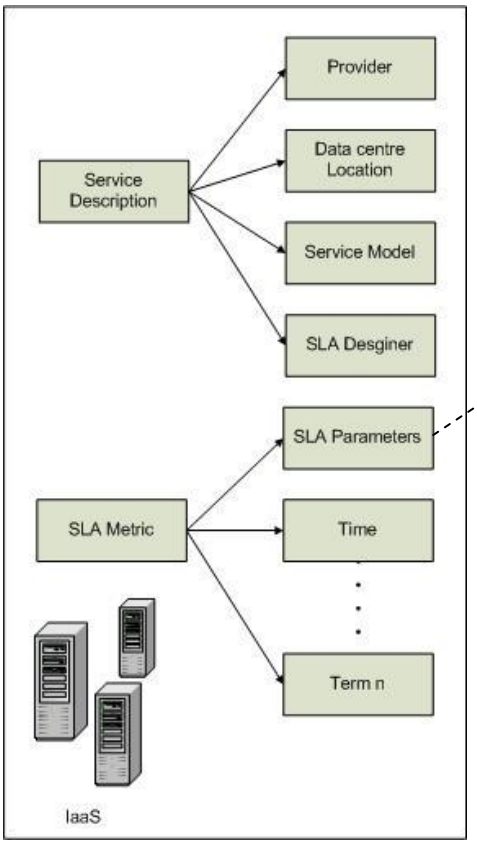
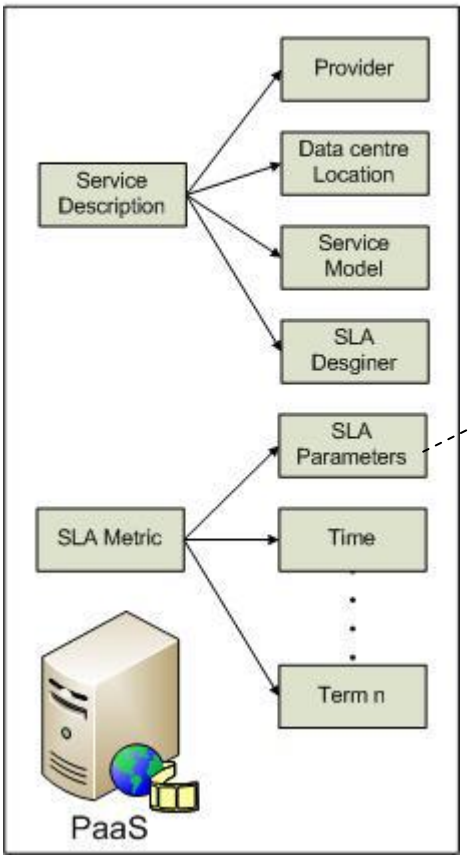


Figure 6.4: SLA metrics for IaaS

Parameter	Description
CPU capacity	CPU speed for VM
Memory size	Cash memory size for VM
Boot time	Time for VM to be ready for use
Storage	Storage size of data for short or long term of contract
Scale up	Maximum of VMs for one user
Scale down	Minimum number of VMs for one user
Scale up time	Time to increase a specific number of VMs
Scale down time	Time to decrease a specific number of VMs
Auto scaling	Boolean value for auto-scaling feature
Max number can be configured on physical server	Maximum number of VMs that can be run on individual server
Availability	Uptime of service in specific time
Response time	Time to complete and receive the process

6.5.2 SLA Metrics for PaaS

Platform as a Service is a type of cloud computing that provides all the requirements needed to support application developers in developing, evaluating, and delivering applications and software for end users [4]. So, in this case, developers using PaaS do not need to download tools or configure hardware to complete the developing tasks. For SLA metrics related to PaaS, I define the main parameters that can be used as the basic criteria when developers want to negotiate with PaaS providers. Figure 6.5 shows the PaaS SLA metrics.

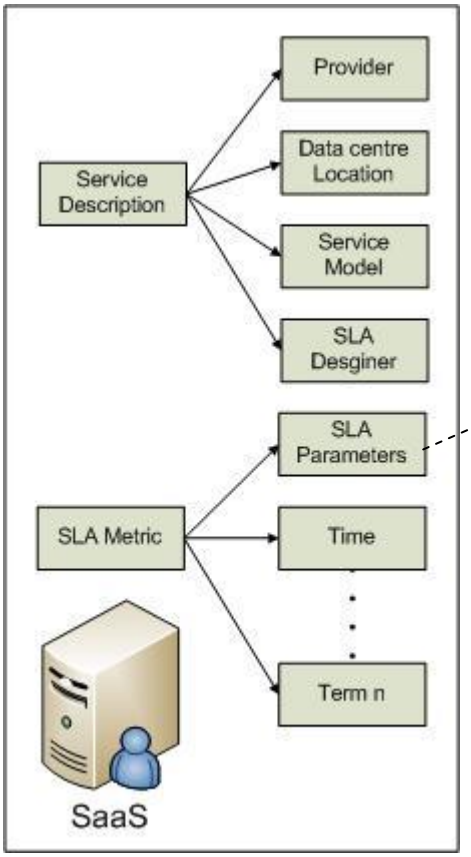


Parameter	Description
Integration	Integration with e-services and other platforms
Scalability	Degree of use with large number of online users
Pay as you go billing	Charging based on resources or time of service
Environments of deployment	Supporting offline and cloud systems
Servers	No. of servers provided for application development
Browsers	Firefox, IExplorer…
Number of developers	No. of developers who can access the platform

Figure 6.5: SLA metrics for PaaS

6.5.3 SLA Metrics for SaaS

Software as a Service is a common example of cloud services [5], users of SaaS use applications developed by the cloud service provider in order to minimize the cost of purchasing software licenses and obtain the benefits of cloud architecture features such as security and scalability. Examples of SaaS are mail, calendar, and social web sites provided by Google, Yahoo, and Microsoft. I present the common metrics parameters for SaaS as an example of metrics for this type of cloud service. Figure 6.6 shows the SaaS SLA metrics.



Parameter	Description
Reliability	Ability to keep operating in most cases
Usability	Easy built-in user interfaces
Scalability	Using with individual or large organisations
Availability	Uptime of software for users in specific time
Customizability	Flexible to use with different types of users

Figure 6.6: SLA metrics for SaaS

6.5.4 SLA Metrics for Database as a Service (DaaS)

In this type of cloud service, online users access their data from different geographical locations. In the past few years, online storage providers were unable to maintain large sizes of data because of the lack of huge space in storage disks, network performance, and data management systems. Now, data storage service providers such as S3 by Amazon.com configure large amounts of storage hardware and they are able to manage and serve millions of users efficiently with their method of data transfer and ensuring these data are compatible with various types of applications. The parameters for data storage service metrics are the basic requirements for negotiation with storage providers. Figure 6.7 shows the DaaS SLA metrics.

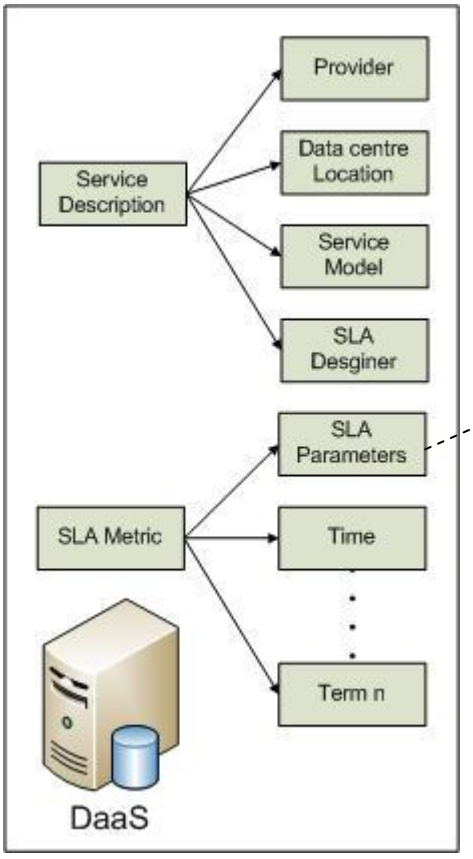


Figure 6.7: SLA metrics for DaaS

Parameter	Description
Geographic location	Availability zones in which data are stored
Scalability	Ability to increase or decrease storage space
Storage space	Number of units of data storage
Storage billing	How the cost of storage is calculated
Security	Cryptography for storage and transferring of data, authentication, and authorization
Privacy	How the data will be stored and transferred
Backup	How and where images of data are stored
Recovery	Ability to recover data in case of disasters or failures
System throughput	Amount of data that can be retrieved from the system in a specific unit of time
Transferring bandwidth	The capacity of communication channels
Data life cycle management	Managing data in data centres, and use of network infrastructure

6.5.5 SLA General Terms

The above section presents the main parameters for the four types of cloud services. However, there are general metrics that can be defined for SLAs with any or all types of cloud users. I present the most important parameters as an example when creating the basic SLA contract between cloud computing users and the corresponding cloud providers. Table 6.1 shows the general SLA metrics.

Table 6.1: General SLA metrics

Term	Description
Monitoring	Who does the monitoring and what monitoring method is used
Billing	Cost of service and how this can be calculated
Security	Issues like cryptography, authentication, and authorization are main requirements for cloud users
Networking	The number of IPs, throughput, and load balancing
Privacy	How the data will be stored and transferred
Support service	Cloud providers should clearly define the methods of help and support
Local and international policies	The policy standards that providers should follow

6.6 CONCLUSION

An effective service level agreement is the key to ensuring that a service provider delivers the agreed terms of services to the cloud consumer. In cloud computing, cloud consumers with a clear definition of SLA parameters and flexible negotiation methods can increase the reliability and trust level of the cloud provider-cloud consumer relationship. In this chapter, the non-functional requirements of cloud consumers are presented and, based on these requirements, the most important criteria for the SLA are defined in order to help cloud users maintain a reliable protocol for negotiation with cloud service providers. The result of this work will be the basic tool for cloud computing to be used with a trust management system, which is proposed in Chapter 8, to help consumers select the most reliable cloud service.

REFERENCES

[1] S. Ron and P. Aliko, *"Service level agreements"*, Internet NG. Internet NG project (1999-2001) http://ing. ctit. utwente. nl/WU2, available at http://ing.ctit.utwente.nl/WU2/, 2001.

[2] L. Taher, H. El Khatib, and R. Basha, *"A framework and QoS matchmaking algorithm for dynamic selection"*, International Conference on Innovations in Information Technology (IIT'05), pp. 26-28, Dubai, UAE, 2005.

[3] E. Chang, T. S. Dillon, T. Dillon, and F. K. Hussain, "*Trust and reputation for service-oriented environments: Technologies for building business intelligence and consumer confidence",* Wiley, 2006.

[4] D. Hilley, *"Cloud Computing: A Taxonomy of platform and infrastructure-level offerings"*, Georgia Institute of Technology, Accessed on April 2009, http://www.cercs.gatech.edu/tech-reports/tr2009/git-cercs-09-13.pdf, 2009.

[5] J. Muller, J. Kruger, S. Enderlein, M. Helmich, and A. Zeier, *"Customizing Enterprise Software as a Service Applications: Back-End Extension in a Multi-tenancy Environment"*, Proceedings of the 11th International Conference on Enterprise Information Systems (ICEIS), Lecture Notes in Business Information Processing, vol. 24. Springer, pp. 66-77, 2009.

CHAPTER 7 – PERFORMANCE MEASURMENTS FOR CLOUD SERVICES

7.1 INTRODUCTION

The ease of use and the ability to scale computing resources on demand has made cloud computing very popular these days. Due to these advantages, users are now able to rent virtual resources from commercial cloud infrastructures with the help of several vendors. Examples of such cloud providers include Amazon and Microsoft Azure. Modelling and determining the performance of cloud providers remains a critical research challenge issue with this new technology. The major reasons for this are the different levels of performance and evaluation of services provided by cloud providers. This chapter of my thesis reviews, from a different standpoint, one of the most widely-used cloud infrastructure services which is EC2 from Amazon. Differences in performance of virtual instances, input output and data transferral are measured by means of experiments. Cloud computing can also have an impact on applications that use networking infrastructure and storage resources. To ascertain the impact on such applications, multiple experiments were conducted. Data was collected on different days, and it was compared with the results that were obtained from local experiments. The scientific results indicated that the VMs' performances differ substantially and quite often fall into two segments that have a large discrepancy in performance. An interesting result was that the two segments of performance obtained corresponded to different virtual system types which were provided by Amazon. Moreover, results were also analysed by taking into consideration different zones of services location, points in time, and the locations themselves. The analysis of results also determined that the selection of a service location also influences the level of the performance. One of the significant conclusions of the work conducted is that the differences between VMs are currently very high. Thus, performance measurement experiments need to be performed with care. Some recommendations are also provided to the users.

The structure of this chapter is as follows. In the second section, the survey presents the existing methodologies that have been proposed for performance evaluation of cloud services. A detailed overview of the Amazon EC2 cloud has been provided in sections 3 and

4. Moreover, the different benchmarks used in the experiments will be presented in section 5. The next section discusses the results in detail. Finally, section 7 concludes this chapter.

7.2 PERFORMANCE MEASUREMENTS

A new paradigm of technology which enables users to access and configure a huge amount of computing resources is known as ‘cloud computing architecture’. It is very easy to use and can be implemented using a number of resources. For these reasons, it has gained popularity. Hence, major providers like Salesforce and Amazon are already offering this new technology. Cloud computing is very attractive to a wide range of users. This includes both researchers and government organizations because the maintenance of the infrastructure is the responsibility of the cloud providers. Hence, cloud providers have paid much attention to the model. However, some VMs may attain a magnitude of order worse than the other existing VMs [1]. This fact can indeed considerably influence the performance of other applications. This can be illustrated with an example. It can be inferred that the performance of EC2 differs to a great extent.

There are several reasons for these inconsistencies which include the conflict for the VMs such as the response time which is also one of the prime reasons for this variability in performance. This irregularity in performance is in fact one of the major issues of cloud computing which many users face and is also considered as one of the major barriers to the success of cloud computing [1]. For example, for any application, developers expect similar performance and at any time. This is apart from any consideration of the existing workload present on the cloud. Also, it is quite important to decision makers because the results are repeated time and time again. Another good example is that clouds depend on SLAs like a grid computing system that has to be constructed within a stipulated time frame. Hence, enterprises also expect cloud providers to offer a quality of service guarantee. Thus, it is critical that cloud providers ensure that a Service Level Agreement is based on performance features, which include storage size as well as the security level. However, the general tendency of cloud providers is to create their SLAs on the basis of the trust level of the services they are offering [2-4].

Thus, currently there is an obvious lack of users since they have to deal with the issue of the irregularity of results. They need to deal with these so that they are better able to understand the difference in the performance of the cloud.

This chapter will focus on the issues mentioned above and will also conduct an in-depth assessment of the performance level of the virtual machines of EC2, as it is one of the best cloud networks known to date. The major contributions that I make in this chapter are as follows. First, I conducted the experiments in different scenarios to test a single VM. This provided an estimation of the difference in performance of a single virtual machine. Also, I used various types of VMs. This provided an estimation of the difference level of performance using multiple VMs. To test the performance in different locations, I conducted different experiments using two locations that gave an estimation of the different levels of performance of cloud providers' infrastructures which are in different locations. The second contribution is the analysis of the results which have been obtained, focusing on the statistical concepts measuring differences in cloud service performance. In this case, I increased the number of VMs and measure the difference in performance levels in order to compare these with low level of performance in fewer VMs. Also, I compared the performance in these scenarios with real applications of cloud computing. The third contribution is my analysis of results and defining the methodology to divide the performance level into two segments. I use various factors for this objective, but the focus will be on virtual machines provided by Amazon. Some recommendations have also been provided to the users so as to reduce the difference in the performance.

It is expected that the study conducted will have a great impact on practice for three main reasons. They are:

- The decision makers obtain a good understanding of the results which have been acquired by conducting experiments using this well-known cloud provider.
- A deeper understanding of the service level agreements is obtained from Amazon EC2 which allows us to see what is being offered to users.
- Suggestions have also been provided to show what can be done to reduce variations in performance.

The above three factors contribute to the significance of the study. This study is currently the first and only one of its type on Amazon EC2, as discussed below.

One of the focuses of several researches is the new method of distributed services. Cloud computing is a hot topic for many researchers. Hence, many evaluations of cloud services have been proposed with different objectives and motivations. According to Leavitt [5], one of the major barriers for cloud computing is the irregularity in the performance of cloud services. Several reasons for this were discovered including the fact that some technologies which are used to develop data centres are too brief to be in scalable and sharing forms. Zhang et al. [6] produced a model of general cloud computing with the aim of distributing cloud technologies and services into different levels which resulted in an understanding of how the technologies can be used and also interchanged. Buyya et al. [7] stated that there are few general benchmarks and these are not enough to gain an understanding of the new services provided by cloud computing as fixed rules are required by them, and concepts such as the reliability to the privacy of sensitive data which are very important to cloud computing, are not considered. Further, Wang et al. [8] also discussed several guarantees and finding the best service related to service level agreements for the cloud platforms. Wang [9] placed more weight on how to secure resources for cloud providers and considered that there is risk involved in sharing the virtual infrastructure among many unknown and possibly untrustworthy people. A comparison of the technology used in cloud computing and storage resources with the help of a few elements was proposed by Doelitzscher et al. [10]. The economics and the monitoring of service operations related to the technology and the different architectures of databases were given by Zhang et al. [11]. The problem of difference in performance was mentioned in their study but further evaluation was not done. Other authors [12, 13] discussed the different cloud services provided by EC2 in relation to service price and performance, but there was no discussion or evaluation of how different performance results can affect customers' applications. Li et al. [14] illustrated the irregularities in performance on the cloud network services from EC2 but focused only on the applications level aspect, that is, the performance of databases resources. Hence, a detailed overview of the cause of these differences in performance has not been discussed. Many new projects have been developed to evaluate the performance of the platforms of cloud computing. Examples include [15] and [16] which are already monitoring the performance of the cloud platforms. However, the performance irregularities which can be encountered have not been evaluated by any researcher; nor have suggestions been given to enable users to understand how this variability can be reduced. A number of studies have also been conducted to compare the performance of the cloud computing with that of the normal models of performance so that the applicability of cloud computing on web-based

applications can be tested [16, 17]. However, these studies focus only on the runtime taken overall and not on the variability of the performance.

Several studies on the scalability of virtual machines already exist. Most of these studies considered the measurement of performance metrics on the local machines. The background loads of tested machines are controlled to compare the results of performance with a different scale of loads. To the best of my knowledge, to date, no such methodology has been developed to study the performance of cloud providers by considering the use of different metrics of performance. For example, Evangelinos and Hill [13] evaluated the performance of Amazon EC2 to host High Performance Computing (HPC). They use a 32-bit architecture for only two types of Amazon instances. Jureta and Herssens [18] propose a model called Quality-Value-Dependency-Priority (QVDP) which has three functions: specifying the quality level, determining the dependency value, and ranking the quality priority. These functions consider the quality of services from the customers' perspective. However, the performance issues related to cloud resources are not discussed and details are missing regarding the correlation of the quality model with the costing model of services. Cherkasova and Gardner [19] use a performance benchmark to analyse the scalability of disk storage and CPU capacity with Xen Virtual Machine Monitors. They measure the performance parameters of visualization infrastructure that are already deployed in most data centres. But they do not measure the scalability of cloud providers using the visualization resources. However, in contrast, my proposed work profiles the performance of virtualization resources that are already running on the infrastructure of cloud providers such Amazon EC2 services.

7.3 SELECTING THE CLOUD PROVIDER

The Amazon EC2 platform is an example of IaaS and DaaS cloud services. Amazon EC2 is one of the biggest providers for large numbers of cloud users who rent cloud resources as a local operating system without spending too much on operating and managing a computing infrastructure. In the context of Platform as a Service, the Google AppEngine provides a developing environment with more usability and flexibility with many programming tool features that can be run on most internet browsers. Salesforce is an example of the most trusted leader of SaaS that provides cloud-based CRM software. In my research, I apply the proposed methodology for performance measurements on an Amazon platform. I focus on evaluating the performance of two types of virtual machines which are deployed on Amazon

EC2. The other three types of cloud services are not considered in my performance measurement methodology. Figure 7.1 shows the screenshot of the Amazon EC2 website.

Figure 7.1: Screenshot of Amazon EC2 platform

7.4 THE CLOUD COMPUTING MODEL OF EC2

The infrastructure produced by Amazon cloud services was initially not supposed to be a cloud services platform. Its main idea was to increase the use of resources available for customers at the peak period of time. EC2 was released to solve the problem of non-scalable systems of offering services. When it came into existence, it became the first commercial public cloud offering IaaS. And today, it allows Amazon to offer a wide range of services apart from CPU resources and storage resources. Amazon EC2 is extremely popular among both distributed service providers and enterprises which are in search of instant and scalable resources for their business needs. This is also one of the prime reasons that the solution proposed in this thesis focuses on the study of this type of cloud service and not others. A dynamically resizable computational capacity is provided to the users of cloud computing. This helps to change the economics of the platform of computing because the users are required to pay only for the amount of resources used for different applications. It is also

known as the ‘ pay-as-you-go’ model. The resources of Amazon EC2 have been implemented with Windows and Linux operating systems using virtualization concepts. These virtual machines are known as instances. To frame it, a true virtual environment is provided which helps the users to use the interface of the Web Service and also rent instances for their use. They can also load them with the help of flexible applications and also take care of, and manage the permissions related to network access. The instances which can be acquired are classified into three types which include standard instances that are good for most of the applications, high memory instances which are good for the throughput-based applications and high CPU applications which are good for CPU-based applications.

Small instances have been considered for the purpose of evaluating the performance as they are the instances with a default size and are multi-purpose. The power of computing is used to classify standard types of instances. It also corresponds to the physical layer of resources. The following are the classifications:

1. The default small instances have around 1.7GB of the main memory and a platform of 32 bits.

2. The next is the large instance which is implemented with a memory of 7.5GB, and platform of 64 bits.

3. The last is the extra-large instance which has a 15GB of memory, and platform of 64 bits.

4. A CPU capacity of a 1.0 GHz processor was found to be equivalent to one EC2 compute unit [20]. However, there are many models which exist in the cloud computing market and it is not clear as to what CPU performance can in each instance help to obtain. There are few resources like CPU, memory and storage which are committed to a particular type of instance, but there are still resources like storage and networking which are shared by multiple resources. If on a physical machine each instance tries to obtain the maximum amount of the shared resources available, each will receive an equal part or share of that particular resource. However, if it is found that a shared resource is not utilized adequately, then the instance may be able to acquire a larger part of the resource. The overall performance of the resources which are shared also depends on the type of the instance which generally contains an indicator that is either moderate or high. This indicator has an effect on the allocation

of these resources. Moreover, currently there are three different physical locations two of which are in the United States and one is in Ireland, with plans to extend to other locations. Each of these locations has zones of different availability and each is autonomous which is beneficial in cases of shutdown [20].

7.5 EVALUATION APPROACH

In this section, the methodology of performance measurements of Amazon EC2 is presented. In order to measure the performance of cloud providers, the traditional method is to run identified applications on a specific type of cloud platform. Examples of these applications include encryption, image processing, data retrieving applications. Although these applications can show the general performance of cloud providers, methods for evaluating a low level of cloud infrastructure are missing in the literature. In this section of the thesis, I present benchmarks that I use to evaluate the performance of an IaaS cloud service which is EC2. Then, I present the approach used to conduct the experiments.

7.5.1 Measurement Benchmark

The performance of the cloud provider needs to be evaluated. To do so, classic cloud computing applications like Java-based applications can be run, or other storage applications can be used in order to evaluate the general performance of given type of cloud services. These applications provide a good idea about the cloud platform's general performance; however, a deeper insight into the performance requires research. Hence, the focus was mainly concentrated on low level benchmarks which helped to measure the performance of each component present in the cloud data centres. Apart from understanding the results of performance, the measure of this performance also helps the users to better understand the working of the new application. The relationship between results and web-based applications must be established and its impact on the size of the virtual machines, different levels of quality as well as the Java applications had to be analyzed. The following sections discuss the different components of the infrastructure, the aspects that the research focuses on, and the performance measurements methodology used in this thesis.

There must be a clear definition of the research components whose performance needs to be measured and the way in which this would be measured. In order to conduct the proposed

methodology, the following factors will be measured which are supposed to have a considerable influence on the performance of the new applications. First, there is the VM starting time. This is important because it helps to analyze how the clouds manage to quickly scale up during peak performance. The second factor, the CPU capacity, is the most important factor for IaaS users involved in processing. The third factor is data transferring which is important because instances are required by the cloud feature to store transitional results and also in cases where the transferred data was not processed by the main memory or default system. The fourth factor is the speed of the memory which is important as both time and data need to be fetched for some applications such as database applications. The last factor is the bandwidth of the network between the virtual machines of the cloud platform.

In the cloud computing environment, there are three important issues which may influence the performance. The three issues are: the different levels of performance of the small and large VMs, the location of cloud data centre, and the availability of different zones. This part of my thesis explores these three aspects in detail.

In this section, I discuss in detail the different benchmarks used in the study. These benchmarks have been used to measure the performance of each factor. The measure of the VM starting time from the time a request for a VM to the time the VM that was requested is available. The state of any starting VM is checked after each time period of 10 seconds and this check is stopped once the VM is ready to use. The multiplication Java-based benchmark utility is used to measure the performance of the next factor, that is the CPU [21], which is used for measuring CPU as well as memory performance. The calculation of different multiplication metrics is executed which helps to provide the score of CPU performance in a single case. To properly evaluate the multicore VMs, I evaluate the performance of two concurrent processes which the CPU is executing. The benchmark used for testing CPU performance can also be used to evaluate the performance of the memory [21]. Write and read tasks of memory operations from one memory to another are executed to obtain the score of the performance. It is also known as a single case of memory test performance score. Further, to measure the performance of the data transferring, I use the Ubench benchmark which is a storage and data transferring benchmark [22]. The results obtained were based on different aspects of data transferring performance.

For bandwidth, the network micro benchmark was used [23] to measure the performance of the network. It is a modern technique for measuring the network's bandwidth performance.

The benchmarks were run at different times each day for a period of 15 minutes. They were run on both small and large VMs. Long-term measurements were taken because the results of the performance change greatly over time. This extended testing time also allowed us to analyze the system performance which is more meaningful. The system which is tested is Amazon EC2 for a period of 20 days, but any additional information could not be obtained because new patterns could not be observed and hence have not been presented in this study. All VMs were shut after 30 minutes which allowed the EC2 platform to produce a new VM just before all the benchmarks were run. The prime reason for this was to distribute the tests over different VMs and thereby to obtain a realistic and holistic measure of the benchmarks used in the study. Also, to prevent the results of the benchmark from impacting on each other, all the benchmarks were run sequentially which ensured that only one benchmark ran at a particular time. A problem was that when running a single benchmark at a time, the benchmark could have lasted for more than 20 minutes. To run the micro benchmark, two idle VMs are required. Hence, before running it, the two VMs were synchronized. Along with this, since the two VMs were not available in the same availability zone, one thing which is quite likely to be different is the network bandwidth. For this reason, different types of experiments were run when the VMs were in the same availability zone and also when they were not.

7.5.2 Set-up of the Experiment

Experiments were run on the EC2 platform of Amazon and for this, small as well as large standard VMs were used from locations in both the United States and Ireland. The number of VMs has been increased in a further section. To understand the hardware of these instances, Section 7.3 can be referred to once again. The Windows operating system was used for both types of VMs. An image of Amazon machines was obtained for each instance for each location which also included the required benchmark code. Standard VMs and local disk were used when running the micro benchmark. All the results obtained from them were stored on Microsoft Excel sheets in the local machine. Further, to compare the results obtained from Amazon EC2 with the baseline, all the benchmarks were run on the local machine which had physical resources. The configuration of the local machine is: one 2.67 GHz Windows 7, 64-bit platform, RAM of 4GB, and 160 GB hard disk. There was full control of this machine and no extra workload was imposed by the other applications during the experimentation process. Hence, it can also be considered as the best case which in turn

was considered as the baseline. Default settings were used for all the experiments. The performance of the compiler which also affects the micro benchmark performance was used on all the instances of EC2 platform and also the physical resources of the VMs. Amazon EC2 is a cloud IaaS provider network which is used all over the world, so because of the difference in time zones, no local time is specified. Hence, the decision was made to use EC2 as time for coordination and also for presenting the results.

In this thesis, I make use of statistical measures to evaluate the variability in performance. For this purpose, I used the standard deviation. Standard deviation is a statistical measure which is widely used as a measure in multi-domains but it is also difficult to compare it with different types of measurements. To reframe it, the standard deviation helps to determine the extent of the difference in performance when compared to the mean value. In addition to this, the proposed methodology also represents a comparison of various measurement scales used. Due to the nature of the problem discussed in Chapter 3, the term standard deviation (σ) is used. The standard deviation (σ) of performance results has been calculated over a sample of results for which the standard deviation of the sample itself is considered. The formula for the standard deviation (σ) is as follows:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} (Mr_i - Mmr)^2}{n-1}} \qquad (1)$$

Where n is the sample of measurements. (Mr_i) is the result which has been measured and (Mmr) is the mean of all the results obtained. The standard deviation of differences of performance enables a comparison to be made of the degree of difference of one data sample among the total samples. This can be done even if the mean is differentiated from each sample.

7.6 EXPERIMENT RESULTS AND ANALYSIS

The experiments were conducted with one crucial thing in mind, which was to measure the difference in the performance of the Amazon EC2 platform, and then analyze its impact on

real-world applications. In order to discover this, the factors mentioned in Section 7.4 were benchmarked and results were obtained which have been tabulated as the baseline in Table 7.1.

Table 7.1: Performance measurements for Amazon EC2 VMs

Small	**Medium (H-CPU)**	**Large**	**Extra Large**	**Extra Large (H-CPU)**	**Local Machine**
656	375	110	125	125	360
734	375	125	172	125	360
844	375	109	124	125	359
650	438	172	187	125	360
STDDEV 122.9	STDDEV 23.5	STDDEV 48.9	STDDEV 27.7	STDDEV 7.2	STDDEV 1.1
Average 769.3	Average 383.2	Average 126.9	Average 153.8	Average 129.6	Average 359.8

7.6.1 VM CPU Capacity

The multiplication Java-based application was used to determine the performance of the CPU. The results are shown in Figures 7.2 and 7.3. The results indicate that the performance of the two VMs differs to a considerable extent. Two segments have been identified. The first segment ranges from 550 to 780 for all the small types of virtual machines and from 90 to 220 for the large types of virtual machines. It was also found that most of the measurements lie in one of the mentioned segments. The σ was also found to vary to some extent for the VMs. It was higher for the larger type of VMs than for smaller types of VMs. In summary, it was found that the performance of both types of VMs was far less stable than expected.

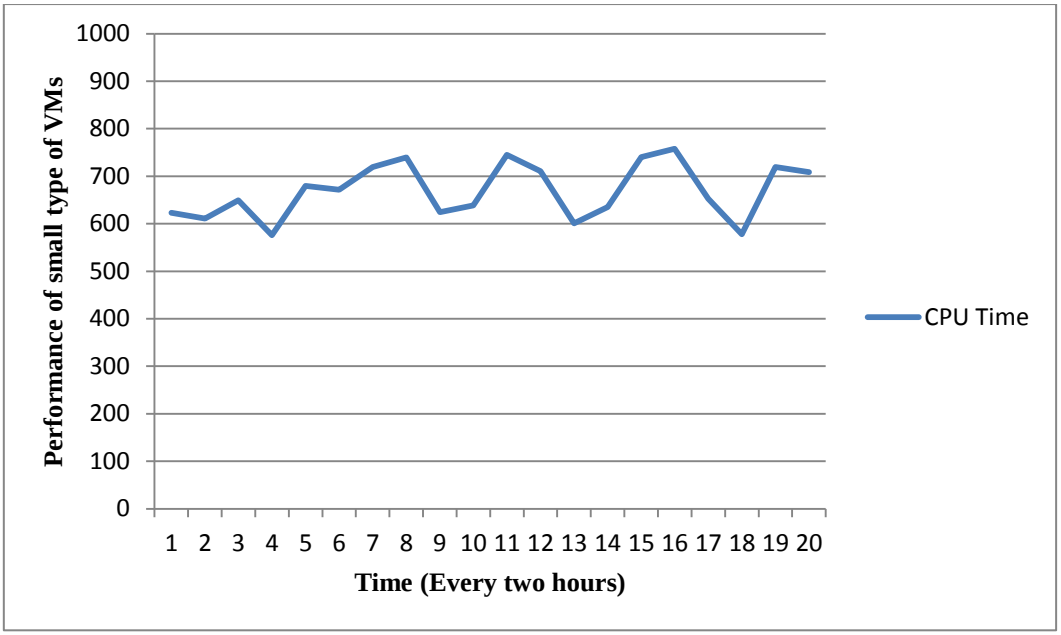


Figure 7.2: Performance of CPU for small VMs

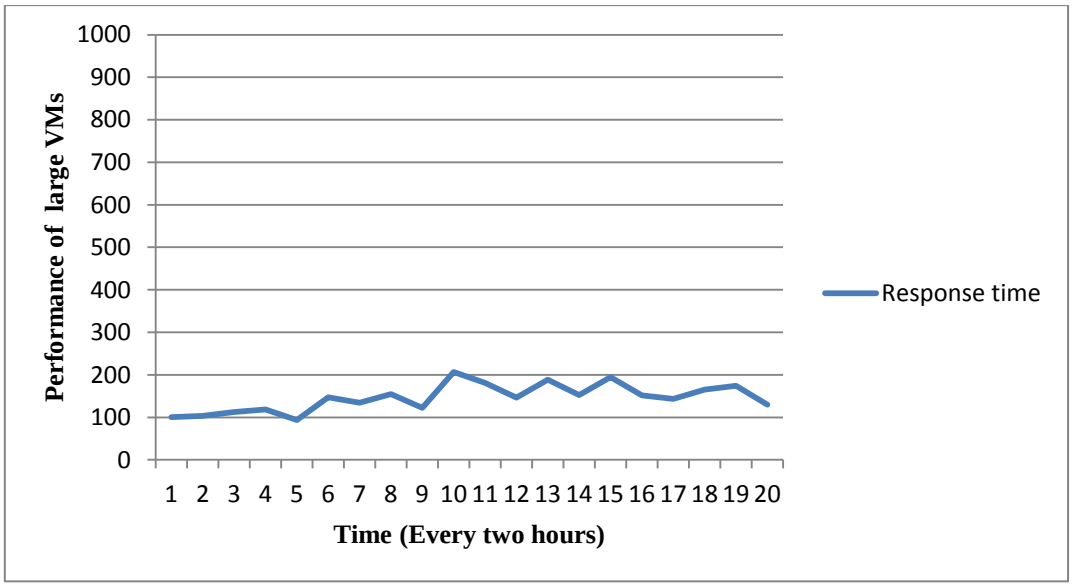


Figure 7.3: Performance of CPU for large VMs

Two segments can be seen. One is very wide and spans most of the domain from 45000 to 70000 KB/s. Along with this, a narrow level can also be seen from 87000 to 97000KB/s. Out of so many, none seems to follow a normal performance level. A likely reason for this could be the effect of the memory size of VMs, being either low or high. I carry out further analysis to determine the actual reason.

As indicated by [24], EC2 contains at least two different types of processors. An additional experiment was conducted to discover the impact of the different processors or systems. To perform this, five types of VMs were initialized and the micro benchmark was run 10 times on them. The results are shown below in Figure 7.4:

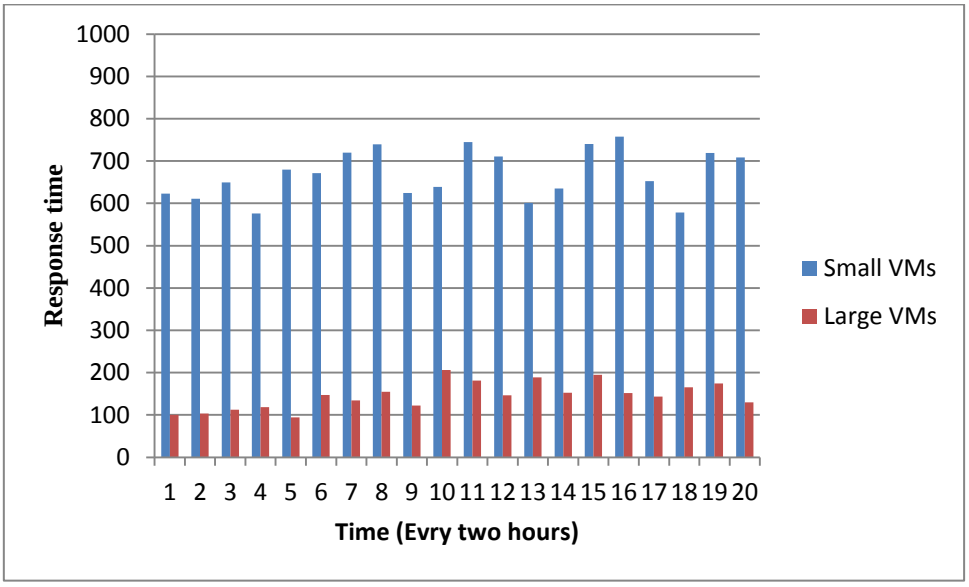


Figure 7.4: Comparing performance of two types of virtual machines

The results show how the performance has been divided into two segments. A similar observation was made in Section 7.5.1. The reason for this is that both VMs are assumed to give the same level of performance. Also, the difference in these segments is much less than the actual difference. During the experiment, different segments of performance of VM memory were also estimated as there was a difference. Storage performance also provided the same kind of results, that is, two segments for the VMs types.

The mean for the performance of the CPU for large VMs is almost four times the performance of smaller VMs. As mentioned previously, a small VM is based on a 32-bit platform and large VM on a 64-bit platform, thereby limiting the comparison. It is also deduced that the VMs' performance lies in the lower segment.

I concluded from the above results that the difference in performance is generally independent of the time when a VM is being used. An in-depth analysis of the same will be done here by looking at different days of the week. The figure below shows the performance for CPU for each week day.

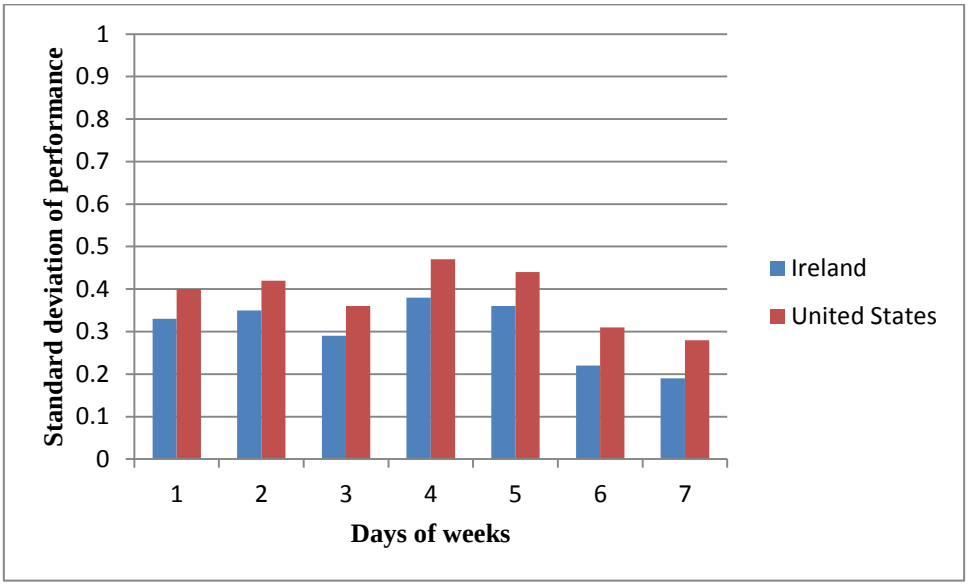


Figure 7.5: Standard deviation of two types of virtual machines

For VMs in the United States zone, a smaller difference in the performance is observed on weekends. For the rest, there is a better performance level. For the Ireland zone, it is non-existent, possibly because users generally run the application during working hours.

7.6.2 Memory Speed of VM

Figure 7.6 shows the results obtained for the memory benchmark. In line with the other benchmarks, this figure depicts two segments of performance for both virtual machine types.

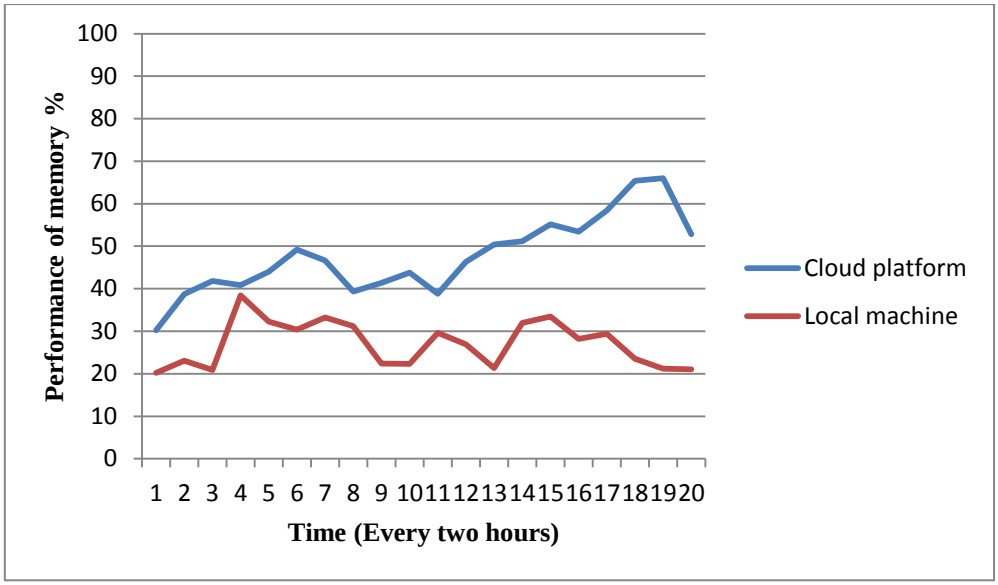


Figure 7.6: Performance of memory

The small type of VMs was found to have a standard deviation less than the large type of VMs. However, the standard deviation for my local experiment was smaller than for both types of VMs on Amazon EC2. This experiment shows that the memory capacity is very high compared to that of my local machine memory units. The situation is more significant for large types of virtual machines. It was also found that the speed of Amazon EC2 was less stable than that of the local machine. The different times to use the resources by large number of customers causes the difference in the capacity of RAM of cloud computing.

7.6.3 Data Transferring Between VMs

Data transferring is measured between virtual machines on Amazon EC2. The results for the write benchmark (between United States and Ireland) yielded almost the same results, as shown in Figures 7.7 and 7.8. It is also evident that the standard deviation of these results is higher than that of the baseline. It was observed that the measurements of data transferring are spread across a wide range of results. Another interesting pattern interpretation is that the measurement for data transferring for large types of VMs differs significantly from the values obtained from the Ireland zone. One reason for this could be different data centre.

It was observed that none of the virtual machines was allocated to the United States zone. Thus, if all the measurements were run from the United States, they would be concentrated in

one segment. This affirms that the availability of zones affects the performance. The same results were obtained for the small VMs and other benchmarks such as network performance. Hence, it is important to specify the availability zone when requesting a VM. The infrastructure installed in each data centres are not in the same capacity. So, the results of the measurements obtained with different values for each zone of availability.

7.6.4 Network Bandwidth

Figures 7.7 and 7.8 shows the results of the network bandwidth obtained. The VMs present in the United States had more differences than the ones present in the Ireland location. The difference in the VMs' performance was large compared to the standard deviation of local machines. Theoretically, it could have been possible because the concept of EC2 is relatively newer in Ireland than it is in the United States resulting in more requests being run over the United States network, thereby sharing more information. However, sufficient information is not available to confirm this theory. Thus, the observed range of obtained measurements is much larger than that of the baseline.

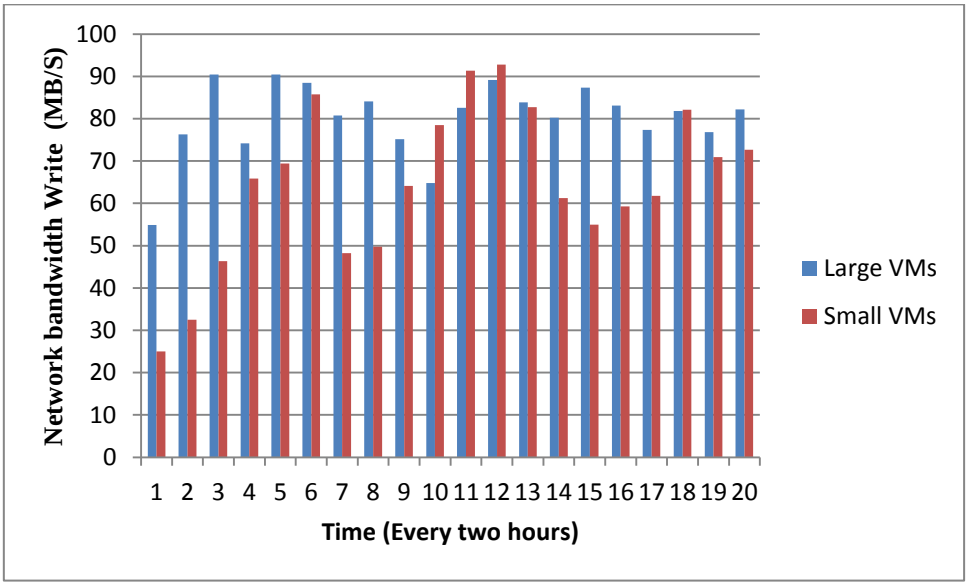


Figure 7.7: Network bandwidth of write benchmark for United States availability zone

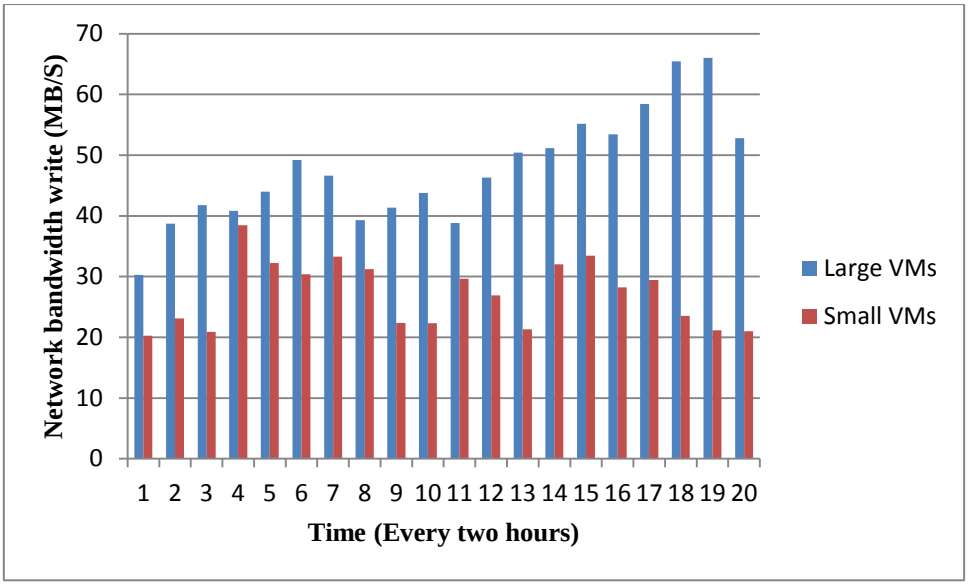


Figure 7.8: Network bandwidth of write benchmark for Ireland availability zone

As mentioned previously, the zone of availability was not considered to be a factor for the experimental set-up, so it received little attention. According to Amazon, each separate availability zone has been implemented so as to be protected from the failures that occur in the other availability zones [24]. I analyze the effect of using the two available zones. The assumption in this part of my thesis is that the same availability zone assigned to two virtual machines running the same network benchmark must be better, and if they are assigned to different zones, then the resultant performance should be less than normal. If this is found to be true, it can be concluded that in an available zone, network units within are better than connection among the units.

For the Ireland zone, I see that most VMs are allocated to the same zones of availability location. However, if the same is analyzed for the United States zone, it is the opposite. The measurements vary to a great extent but the measurements in an availability zone have an average performance of 32 and among different ones, it is 54. Thus, the performance within is better than otherwise. This conclusion was validated with further experiments. It was concluded that the network performance could be improved with the help of an allocation approach which schedules users' requests.

7.7 CONCLUSION

To date, micro benchmarks have been run on both small and large types of virtual machines. Now, two things would be analyzed:

1. The variability in performance discussed in this part of the thesis averages out the consideration of different types of virtual machines.

2. The impact of smaller virtual machines on applications which are using databases most of the time.

Considering performance measurements as a random factor, average performance can be expected to have less impact due to the larger number of samples. Thus, virtual machines were used for experiments, but the important relationships and differences between the types of VMs could not be established.

One important thing to consider is the way that performance irregularity of IaaS resources impacts on the performance of applications. A Java-based application has been used as a benchmark here. This is because it is widely used and hence, the results would interest the users. Also, the application makes use of intense memory, CPU and network.

Cloud computing is a new form of technology, whose infrastructure, developing platform, software, and storage can be delivered as a service in a pay-as-you-use cost model. Intelligent usage of resources in cloud computing may help cloud customers to reduce the large amount of IT investments as well as operational costs. However, for critical business applications and more sensitive information, cloud providers must be selected based on a high level of performance and trustworthiness. To use cloud services, it is very important to understand the performance of the cloud infrastructure provided by clouds. In this chapter, I evaluated the performance of two types of EC2 virtual machines as an example to examine the stability of most types of VMs which are provided by Amazon platform. I demonstrate that the difference in performance of small types of VMs has the best stability. So, as a service level agreement, performance metrics can be used as a good parameter in the agreement. But for large types of VMs, it is important to improve the stability of performance before signing any agreement between cloud provider and user.

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, and I. Stoica, *"Above the clouds: A berkeley view of cloud computing"*,EECS Department, University of California, Berkeley, Tech Rep, UCB/EECS-2009-28, 2009.

[2] G. Wang and T. S. E. Ng, *"The impact of virtualization on network performance of amazon ec2 data center"*, Proceedings of the 29th conference on Information communications, pp.1163-1171, San Diego, California, USA 2010.

[3] D. J. Abadi, *"Data management in the cloud: Limitations and opportunities"*, IEEE Bulletin on Data Engineering, pp. 3-12, 2009.

[4] I. Foster, Y. Zhao, I. Raicu, and S. Lu, *"Cloud computing and grid computing 360-degree compared"*, Grid Computing Environments Workshop, pp. 1–10, 2008.

[5] N. Leavitt, *"Is cloud computing really ready for prime time?"*, Computer, vol.42, pp.15-20, 2009.

[6] L. J. Zhang and Q. Zhou, *"CCOA: Cloud computing open architecture"*, IEEE International Conference on , ICWS, pp. 607-616, 2009.

[7] R. Buyya, R. Ranjan, and R. N. Calheiros, *"Modeling and simulation of scalable cloud computing environments and the cloudsim toolkit: Challenges and opportunities"*, Proceedings of the 7th High Performance Computing and Simulation (HPCS 2009), pp. 1-11, Leipzig, Germany, 2009.

[8] L. Wang, G. Von Laszewski, A. Younge, X. He, M. Kunze, J. Tao, and C. Fu, *"Cloud computing: a perspective study"*, New Generation Computing, vol. 28, pp. 137-146, 2010.

[9] C. Wang, Q. Wang, K. Ren, and W. Lou, *"Privacy-preserving public auditing for data storage security in cloud computing"*, Proceedings of the 29th Conference on Information Communications, INFOCOM'10, pp. 525–533, Piscataway, NJ, USA 2010.

[10] F. Doelitzscher, C. Reich, and A. Sulistio, *"Designing cloud services adhering to government privacy laws"*, Computer and Information Technology (CIT), 2010 IEEE 10th International Conference, pp.930-935, 2010.

[11] X. Zhang and G. Dong, *"A new architecture of online trading platform based on Cloud computing"*, Wearable Computing Systems (APWCS), Asia-Pacific Conference, pp. 32-35, 2010.

[12] A. Mohammed, D. Tharam, W. Chen, and C. Elizabeth, *"Response Time for Cloud Computing Providers"*, Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services (iiWAS '10), ACM, pp. 603-606, 2010.

[13] C. H. Constantinos Evangelinos, *"Cloud Computing for parallel Scientific HPC Applications: Feasibility of Running Coupled Atmosphere-Ocean Climate Models on Amazon's EC2"*, The First Workshop on Cloud Computing and its Applications (CCA'08), vol. 2, pp. 2-34, 2008.

[14] A. Li, X. Yang, S. Kandula, and M. Zhang, "Cloudcmp: Shopping for a cloud made easy,", Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing, pp. 5-5, Boston, MA, 2010.

[15] M. A. Vouk, E. Sills, and P. Dreher, *"Integration of High-Performance Computing into Cloud Computing Services"*,Handbook of Cloud Computing, pp. 255-276, 2010.

[16] A. Iosup, S. Ostermann, N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema, *"Performance analysis of cloud computing services for many-tasks scientific computing"*, IEEE Transactions on Parallel and Distributed Systems, vol. 22, pp. 931-945, 2011.

[17] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, *"CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms"*, Software: Practice and Experience, vol. 41, pp. 23-50, 2011.

[18] I. Jureta, C. Herssens, and S. Faulkner, *"A comprehensive quality model for service-oriented systems"*, Software Quality Journal, vol. 17, pp. 65-98, 2009.

[19] L. Cherkasova and R. Gardner, *"Measuring CPU overhead for I/O processing in the Xen virtual machine monitor"*, Proceedings of The Annual Conference on USENIX Annual Technical Conference, pp. 24-24, Anaheim, CA, 2005.

[20] E. Amazon, *"Amazon Elastic Compute Cloud (Amazon EC2)"*, http://aws.amazon.com/ec2/, Accessed on June 2010.

[21] E. Hu, A. Wellings, and G. Bernat, *"Deriving java virtual machine timing models for portable worst-case execution time analysis"*, On The Move to Meaningful Internet Systems, OTM Workshops, pp. 411-424, 2003.

[22] X. Xu, F. Zhou, J. Wan, and Y. Jiang, *"Quantifying performance properties of virtual machine"*, The International Symposium on Information Science and Engineering, pp. 24-28. 2008.

[23] J. Liu, B. Chandrasekaran, W. Yu, J. Wu, D. Buntinas, S. Kini, P. Wyckoff, and D. K. Panda, *"Micro-benchmark level performance comparison of high-speed cluster interconnects"*, Proceedings of Hot Interconnects 11 (HotI03), pp. 60-65. 2003.

[24] Amazon, "*Amazon Elastic Compute Cloud (Amazon EC2)",* Available: http://aws.amazon.com/ec2, Accessed on October 2011.

CHAPTER 8 – FUZZY-BASED TRUST MODEL FOR CLOUD COMPUTING

8.1 INTRODUCTION

Cloud services are becoming popular in terms of distributed technology because they allow the users to rent well-specified resources of computing, network, and storage infrastructure on demand. Users pay for their usage of services without needing to spend massive amounts on the integration, maintenance, or management of the IT infrastructure. Before interaction occurs between cloud providers and users, the establishment of trust in the cloud relationship is very important in order to minimize the security risk and malicious attacks. The notion of trust involves several dimensions which include: the scalability, availability, security, and usability parameters of IaaS, PaaS, SaaS, and DaaS. Each of these dimensions is characterized by fuzzy aspects and linguistic terms. This chapter develops a model for each of the dimensions of IaaS using fuzzy-set theory. Then the Takagi-Sugeno fuzzy-inference approach is used to develop an overall measure of trust value for the cloud providers. In this research, I will use an IaaS as the main example for the collection of the data and apply the fuzzy model to evaluate the notion of trust in cloud computing. As a part of my outlook, I will extend my trust model to PaaS, DaaS, and SaaS.

8.2 FUZZY INFERENCE SYSTEM

Chang et al.[1] identify number of factors that should be considered when developing trust-based reputation models for service-oriented architecture. To evaluate trust for trusted agents, the trust model should include the following very important characteristics:

a) The recommendation opinion of the third-party agent;
b) The credibility of trustworthiness of the opinion; and
c) The delay which might occur with the passage of time after the last interaction of the third-party agent with the reputation agent.

These factors can be represented by fuzzy sets. Trust is a very important factor in open distributed systems. It is the main concern of all interactions between service providers and

consumers in such a rapidly changing environment. The trust evaluation process is not clear because different users may have vague and different subjective values regarding the individual criteria. This leads to the need for a descriptive methodology that clearly presents the values of trust. Fuzzy logic is an effective technique for solving this problem. The fuzzy logic approach is a better method of modeling and computing human perception. Therefore, this thesis uses this approach to evaluate the trustworthiness of service providers in cloud computing.

In this research, I develop an approach that characterizes the key aspects of the trust relationship between cloud providers and users. Moreover, each of the trust dimensions is represented within a fuzzy framework, and measures for each dimension are developed. In addition, an overall figure for trust value is developed for the cloud providers.

8.3 FUZZY-BASED TRUST MODEL FOR CLOUD COMPUTING

Basing my work on the fuzzy logic technique in the cloud services environment, I aim to solve the problem of uncertainty in the evaluation of trust for cloud providers. This section presents the proposed methodology for evaluating the trustworthiness of cloud users. Also, I discuss how the fuzzy logic approach can be used to solve the problem of estimating the trust level for cloud services.

8.3.1 Problem Definition

The creation of trust and reputation models relies on three main concepts in the development process. These three concepts are: a) building the relationship and trust network; b) storing the trust information; and c) computing the final trust value. My solution for the trust computation problem in cloud computing focuses on the last concept in the development of a trust management model for cloud users which is the computation of the trust value. I use the fuzzy logic approach to produce the final trust value for the cloud service/cloud service provider.

The main problem of producing reliable values for trust is how to define the major factors that affect trust value and also how to use these factors to evaluate the final trust for a given cloud service/cloud service provider. Because of the dynamic nature of trust, the factors

affecting trust should be determined first and subsequently modelled. In this research, I identify the most important factors that can affect the trust evaluation in a cloud community. Based on these factors, I use a fuzzy logic approach to evaluate the final trust of cloud services.

In the existing body of literature on cloud computing, there is no framework by which a cloud service consumer can make an intelligent trust-based decision regarding service selection from a service provider. Given the potential growth of cloud computing and the business implications, it is very important to have such architecture in place. The primary issues which are not investigated in the related literature are:

- The lack of a reliable model for trust and reputation specified for cloud architecture;
- The difficulties faced by cloud users when they want to sign online agreements with cloud providers; there is no clear and reliable method for selecting the most suitable parameters for the SLA contracts;
- The lack of a proposed model to calculate and estimate the cost for each level of the cloud architecture;
- Although trust and reputation systems have been widely proposed and implemented for various types of online services, no such models have been proposed for cloud computing; cloud users also need such systems in order to select the most trustworthy of services that are already being offered by cloud providers.

In my research, I focus on how to evaluate the trusted cloud providers, in such a way that users of cloud can easily understand and start to build a trust-based relationship with service providers.

8.3.2 Trust Factors in Cloud Computing

During the process of designing the SLA model for cloud computing as presented in Chapter 6, various parameters were investigated for cloud services as shown in Figure 8.1.

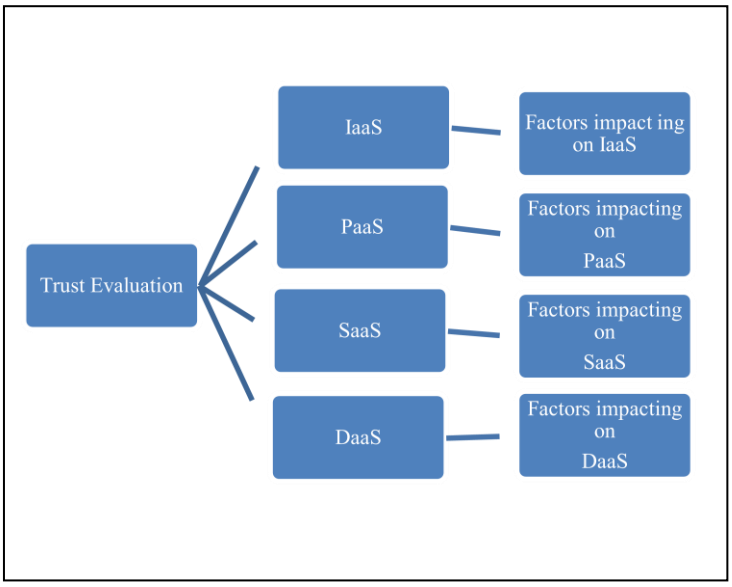


Figure 8.1: Factors that impact on the different cloud services

In the proposed model, I will use the main parameters which were designed for IaaS with additional parameters as the core factors that the cloud computing experts believe will impact on the trust evaluation of IaaS. Table 8.1 lists the factors that impact on the trust evaluation of IaaS-based cloud services.

Table 8.1: Factors impacting on the trust value of IaaS

Final result of trust evaluation	**Factors impact trust**
Degree of trust (T)	Scalability (Sca)
	Availability (Avi)
	Security (Sec)
	Usability (Usa)

Trust evaluation factors include scalability, availability, security, and usability. After identifying the contributing factors, this section describes the inputs and output of the proposed model. This part of the research is usually conducted after investigating the problem domain and establishes more relationships with the cloud computing experts. I established

good relationships with a number of cloud computing experts and end users and were able to determine the most important variables for my model.

8.3.3 Data Collection

One of the most important steps in the development of fuzzy-based control systems is the data set preparation and collection. Therefore, the model with a fuzzy inference approach must be trained with training data that represent the greatest possibilities of application [3]. In this study, I used the data which was collected from cloud computing experts and cloud users. An online-based survey was developed in order to collect more data sets from different locations. The survey with the designed research questions was conducted to collect values for the most important variables which had already been selected to present the trust value in a cloud-based application.

8.3.4 Design of the Fuzzy Trust-Based Model

Fuzzy logic theory is grounded in mathematical constructs in a certain method with fuzziness in order to help make an intelligent decision. By basing my research on the application of fuzzy logic technique to the cloud services environment, I aim to solve the problem of uncertainty in the evaluation of trust for cloud providers. The proposed fuzzy logic method in this thesis uses three fuzzy sets for the input factors and five fuzzy sets for the parameters of output. The three fuzzy sets which are low (L), medium (M) and high (H), are used to characterize the fuzzy value for each input which are scalability, availability, security and usability. The fuzzy sets that represent the output parameters are: very poor (VP), poor (P), good (G), very good (VG), and excellent (E).

The proposed solution uses main two factors to evaluate the degree of trust for cloud users. These two factors are:

1. Trust value (T_v) for cloud service/service provider
2. Credibility of cloud service recommender

To investigate the method of evaluating trust and credibility for cloud users, I use the following equations:

$$T_v = F(Sca, Avi, Sec, Usa) \qquad (1)$$

$$Cr = Q(T_{v,}t) \qquad (2)$$

To obtain the level of security, use the first equation which has four inputs: Sca is the level of scalability factor, Avi is the level of uptime of cloud service, Sec is the level of security, and Usa is the degree of usability of the cloud platform. The second equation is used to obtain the credibility level of the cloud service/service provider. The credibility level is determined based on the trust value (T_v) and the time (t) when the cloud service is used.

The use of three fuzzy sets for the four inputs as shown in Figure 8.2, will generate 241 fuzzy rules. This takes into consideration all possible combinations of inputs. This is a large number of rules and many of these rules are unnecessary when using the Takagi-Sugeno fuzzy technique.

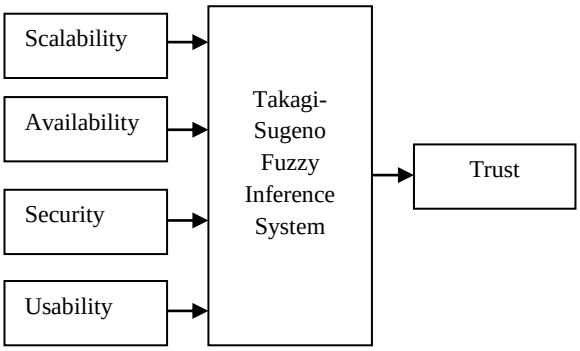


Figure 8.2: Takagi-Sugeno fuzzy inference model

Hence, the neural networks [4] will be used to reduce the number of fuzzy rules. In this way, the proposed model provides a more convenient method of evaluating the trust value of cloud providers. Table 8.2 shows the sample's fuzzy rules for input factors and the assigned values for output. The type of membership function for inputs and output that can be used depends on the nature of the system's attributes. In this research, the bell membership function is used because this is the simplest membership function that can present the input data, and gives a better view when I analyse the results from the experiments.

Table 8.2: Samples of fuzzy rules for trust evaluation of IaaS

IF Sca	AND Avi	AND Sec	AND Usa	Then T
L	L	L	L	VP
M	L	L	M	P
M	M	L	M	G
L	M	M	L	P
M	L	M	M	G
H	L	L	H	P
M	M	M	M	G
H	H	L	H	G
L	H	H	L	G
H	L	H	H	VG
H	M	M	H	G
M	H	M	M	G
H	M	H	H	VG
H	H	H	H	VG

The proposed system will help cloud users to make intelligent decisions using a simple method. Figure 8.3 explains the main process for trust decision-making for cloud providers.

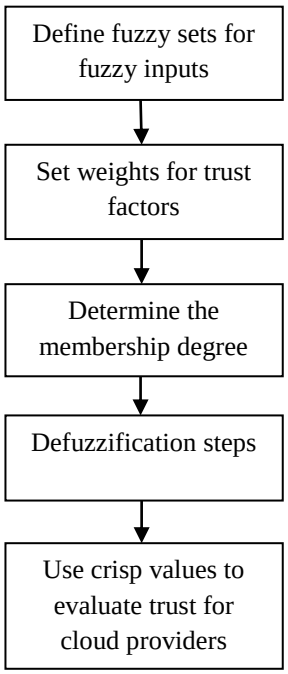


Figure 8.3: Trust decision-making process for cloud providers

The proposed model are explained as follows:

1) The first step of the proposed method is to define the fuzzy sets for all factors and fuzzy sets for the output which is the trust value for cloud providers. Then, the requesters of trustworthiness about a service provider can set the weights for each factor; this step depends on the application of cloud services. If requesters do not like to provide these weights, the proposed fuzzy system will deal with all factors equally. Figure 8.4 describes the workflow of the proposed system.

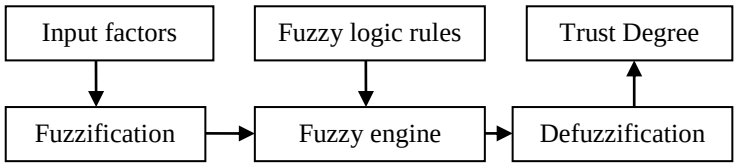


Figure 8.4: Fuzzy model processes

The step with the crisp values is given the fuzzy inference system. For instance, the crisp value of the usability factor can be considered to be 70 out of 100. Crisp values of each factor are converted to fuzzy linguistic terms in order to compute the linguistic term of the final trust value. This step is applied using fuzzy rules which are determined by the training data. To compute the trust level of the cloud service/service providers, fuzzy logic systems define which level of trust value should be produced. For this step, fuzzy membership functions are used to define the final numerical value of trust.

2) The second step is the fuzzification process. In this step of the process, all inputs are assigned to the appropriate degree of the fuzzy sets of input. This process uses a membership function to determine the degree of input to the fuzzy set.
3) Select the fuzzy membership function. Membership function is a function that determines how each of the values in the input range belongs to the input space of the membership value. The membership range is between 0 and 1. Figure 8.5 presents one of the membership functions of the scalability factor.

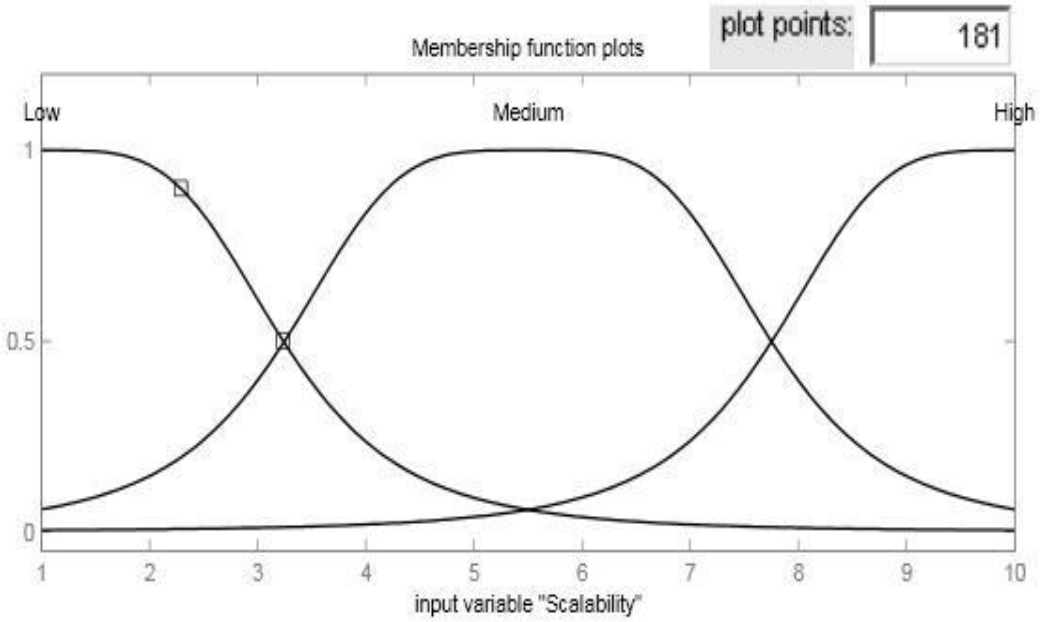


Figure 8.5: Membership function of the scalability factor

4) Design fuzzy rules. In this research, I select the most important rules of the inference system based on the neural networks [4].
5) Takagi-Sugenofuzzy inference engine. The main technique which is used in the proposed model is the Takagi-Sugeno fuzzy inference method. Takagi-Sugeno's method is one of the most popular control approaches which use the fuzzy theory. The Takagi-Sugeno fuzzy inference takes the fuzzy set of inputs to produce a final output value as a crisp value.
6) Defuzzification. This type of fuzzy method uses centroid calculation in the process of defuzzification to produce a single output value.

8.3.5 Quantification of the Corresponding Parameters

To calculate the value of factors affecting trust for cloud service/service providers, I use the following equation for each factor:

$$Factor_x Level = \sum_{i=1}^{n} X_i \qquad \textbf{(3)}$$

Where $Factor_x Level$ is the factor level of trust value affecting the cloud service/service providers, and n is the number of properties of the particular affecting factor.

Security

Security level (SecLevel) is determined by four security properties which are integrity, confidentiality, cryptography, and privacy. The security level can be calculated using the approach proposed by Chang et al.[1] as follows:

$$SecLevel = \frac{\sum_{i=1}^{4} X_i}{\sum_{i=1}^{4} Max\,\mathrm{Pr}\,opertyvalue_i} \qquad (5)$$

Availability

The availability of cloud services can be defined as the time during which the cloud service is available for users divided by the specified interval of time. The following equation defines the main concept of availability for cloud services at given time A_t:

$$A_v = \frac{average_{avt}}{Time_{\mathrm{int}\,er}} \qquad (6)$$

Where A_v is the availability ratio, $average_{avt}$ is the average time for which the cloud service is available to the customer, and $Time_{inter}$ is the specified period of time.

After calculating the SecLevel value, I use fuzzy rules to convert the obtained value using equation (5) to the linguistic fuzzy sets. Table 8.3 provides an example of this process:

Table 8.3: Example of applying membership functions

Property ID	Value of SecLevel	Value of fuzzy set
1	0-28	Low
2	29-45	Medium
3	46-80	High
4	8-100	Very High

8.4 EXPERIMENT

This section shows the validity of my methodology for verifying the proposed trust calculation model for cloud-based online services. In this section, the implementation of the proposed model is provided together with the final results of the experiment. The fuzzy logic toolbox of Matlab is used to design and implement my model. This toolbox includes ready functions and calculation methods to implement more than one type of fuzzy inference systems such as the Mamdani and Takagi-Sugeno inference systems. In my model, I use the Takagi-Sugeno fuzzy method with the bell membership function for inputs and output. Figure 8.6 presents the main model for the fuzzy logic system. I used the FIS editor in Matlab to develop the model. The proposed model was implemented with four input factors: scalability, availability, security, and usability. These four inputs are directed as inputs to the fuzzy inference system implemented with the Takagi-Sugeno method.

After calculating values for all affecting factors, I convert these into a value of fuzzy sets based on the rules that I defined in the previous section. Figure 8.6 shows the fuzzy tool box interface that I used to produce membership functions when the crisp values are converted.

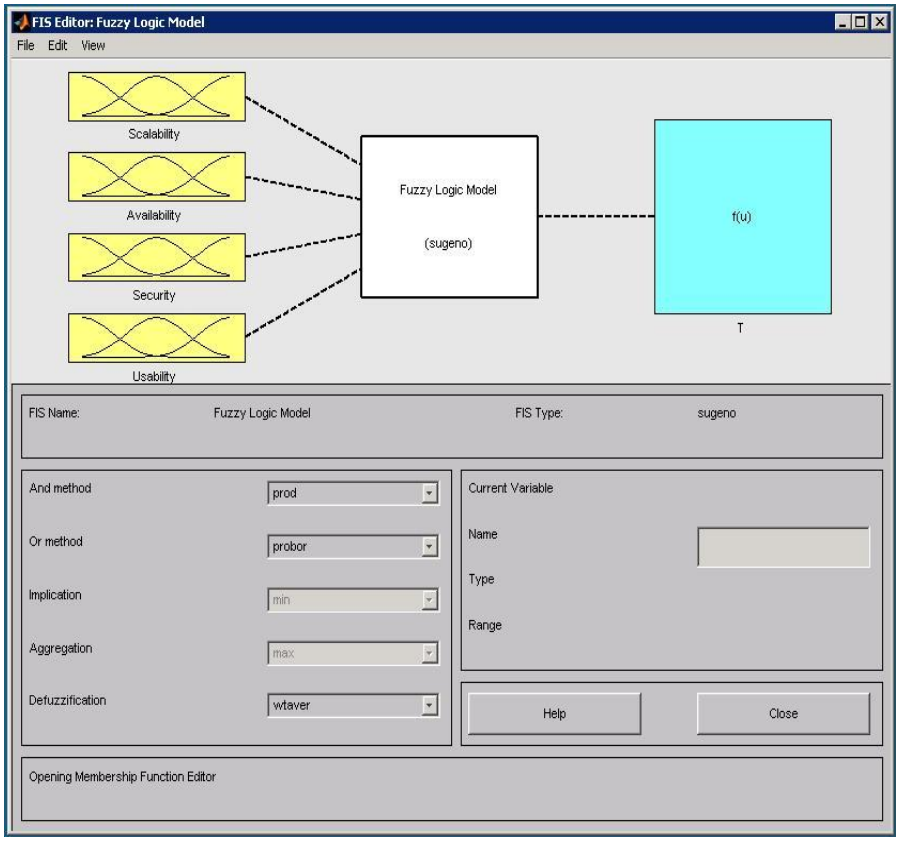


Figure 8.6: FIS editor interface

Using the FIS editor, I trained the system with 54 of the 81 data sets which were collected for this experiment. Figure 8.7 shows the system after the training process.

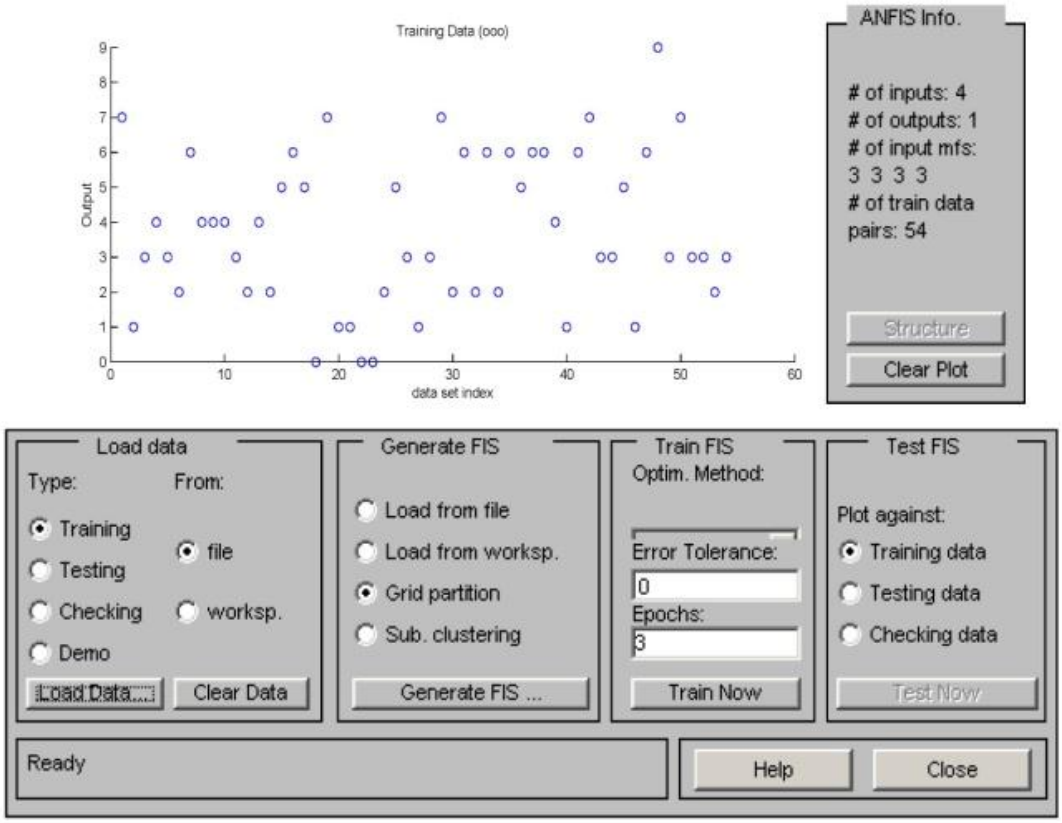


Figure 8.7: Training data sets

In the training process, I undertake two main steps. The first step is to learn the pattern underlying the data. IF-THEN rules and the knowledge from experts are used to learn the system in the first step. In the second step, I use membership functions and select the related rules to learn the parameters of inputs. Figure 8.8 shows the proposed system after the training process using the two training steps.

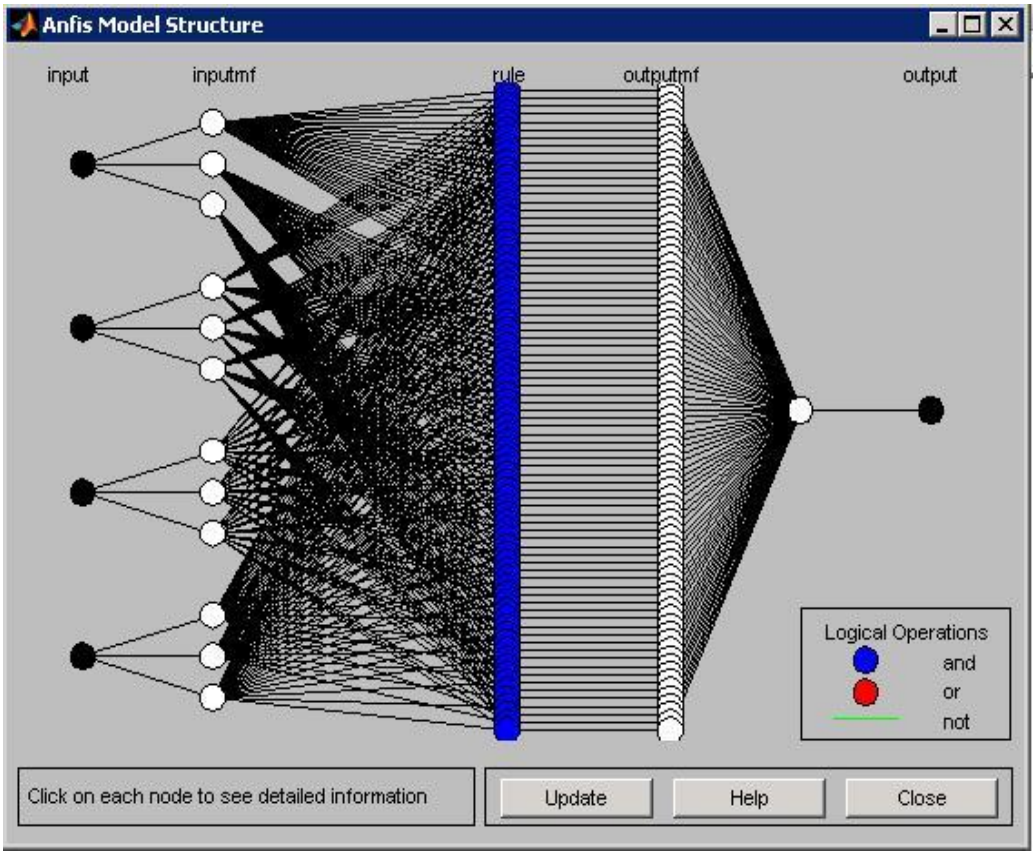


Figure 8.8: Fuzzy inference system after training steps

The security factor has more impact on the trust value of cloud providers. In order to understand the impact level of security on producing total trust in cloud providers, I conduct the following experiment. I test the impact of security in an isolated simulation case. Figure 8.8 shows the different levels of trust in a given cloud provider from the security perspective. In this simulation, other factors such as availability have been fixed as constant values and only the security parameter is varied. Using fuzzy logic sets, I fix those values at low, medium, and high.

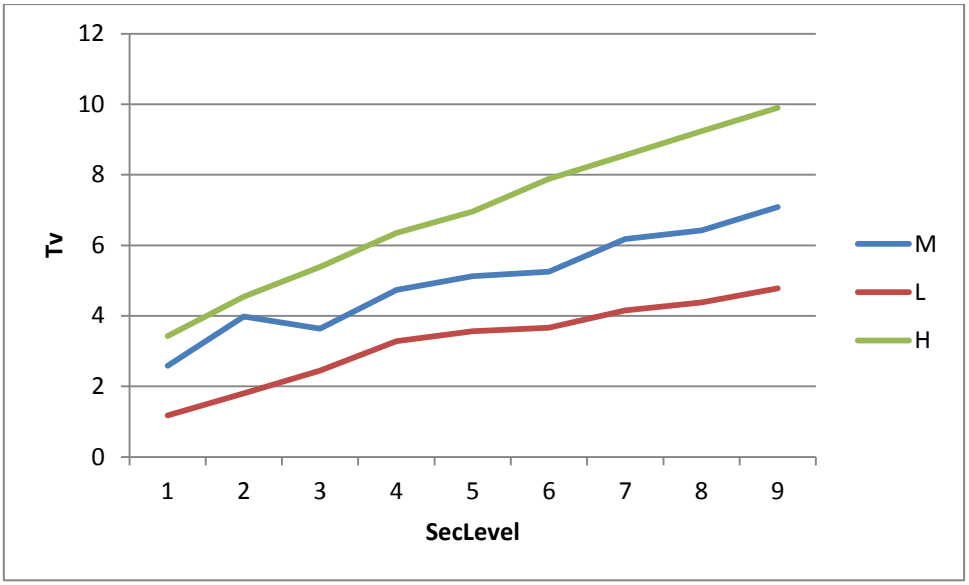


Figure 8.9: Analysis of Scalability factor versus Trust Level

I observe the trust values obtained for the high values of the security factor are higher than the other values in the low and medium range (figure 8.9). This observation confirms that security is the most important factor when cloud customers consider using the cloud platforms of different cloud services.

8.5 CONCLUSION

In this chapter, a trust evaluation scheme based on the fuzzy logic system is described. The proposed scheme enables cloud users to evaluate the trustworthiness of cloud service providers when creating or shifting their distributed systems to cloud data centres. My evaluation method uses the cloud-based application with certain factors such as security and availability as an example of a cloud-based service. I believe that the proposed model provides a valid method, since the results obtained from the experiments which are presented in this chapter are close to the model output using the real data sets. This work is the first and only work of its type in the existing literature. This fuzzy-based model can be extended for use with additional input factors. Moreover, my model can also be extended to various web-based applications such as e-commerce etc.

REFERENCES

[1] E. Chang, T. S. Dillon, T. Dillon, and F. K. Hussain, "*Trust and reputation for service-oriented environments: Technologies for building business intelligence and consumer confidence",* Wiley, 2006.

[2] A. Mohammed, D. Tharam, W. Chen, and C. Elizabeth, *"Response Time for Cloud Computing Providers"*, Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services (iiWAS '10), ACM, pp. 603-606, 2010.

[3] C. Yin, L. Rosendahl, and Z. Luo, *"Methods to improve prediction performance of ANN models"*, Simulation Modelling Practice and Theory, vol. 11, pp. 211-222, 2003.

[4] S. Sestito and T. Dillon, *"Automated knowledge acquisition"*, Prentice Hall, Upper Saddle River, 1994.

CHAPTER 9 – CONCLUSION AND FUTURE WORK

9.1 INTRODUCTION

This thesis presented definitions of service level agreements with consideration given to performance metrics and quality of service criteria of cloud services. The proposed definitions of SLA are provided for use by the different types of cloud users. In order to help cloud users include the most relevant factors of performance and quality of cloud services in SLAs, this thesis also presented a methodology for assessing and measuring the most relevant metrics of performance in different cloud platforms. I provided the solution to help cloud users choose appropriate criteria in relation to the performance of cloud services when they start to negotiate with cloud providers. Finally, I developed a dynamic pricing scheme for cloud services to be used with an SLA framework for developing a trust model for cloud users. The trust model for a cloud relationship provides a reliable way of selecting cloud services with the required level of privacy and security.

Based on the discussion of the literature in Chapter 2, and the description of research contributions in previous chapters, I conclude this thesis and discuss the future directions as follows:

Section 9.2 of this chapter concludes my thesis by providing a summary of thesis contributions, findings and results. In Section 9.3, I discuss the directions of future work in continuing the research in the domain of cloud computing.

9.2 CONTRIBUTION OF THE THESIS TO THE EXISTING BODY OF LITERATURE

In this thesis, I proposed four solutions to improve the reliability and security of providing and using cloud computing. These four solutions, which are discussed in Chapters 5-8, are listed below:

1. Development of a dynamic pricing scheme for cloud services: the service consumers using cloud computing are willing to pay as they use, so an annual billing period or even monthly periods are not suitable for cloud computing. A cost calculation method for resource reservation must be correlated to the method of proposing the price of service in order to maximize cloud service profit and increase the customer demand. I propose a dynamic pricing scheme for cloud computing architecture in order to satisfy consumer requirements and provide an acceptable level of revenue for cloud providers.

2. Development of a methodology for SLA of cloud platforms: the SLA parameters are specified by metrics. These metrics define how cloud service parameters can be measured and specify the values of measurable parameters. In the cloud computing architecture, there are four types of services which providers can offer to consumers. These services are: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), and Database as a Service (DaaS). I proposed SLA metrics for cloud computing which consider these four types of services.

3. Development of a methodology for performance and measurement for cloud services: Performance evaluation is a very important factor in open distributed systems. It is the main concern of all interactions between service providers and consumers in such a rapidly changing environment. The performance evaluation process may not be clear and easy for some users because it has vague and diverse subjective values. Hence, I propose a methodology to present the values of performance metrics in a clear way. Real-world experiments were conducted to measure the performance of Amazon EC2 with different metrics of performance.

4. Development of a trust and reputation model for cloud services: In the existing body of literature on cloud computing, there is no framework whereby a cloud service consumer can make an intelligent trust-based decision regarding service selection from a service provider. Given the potential growth of cloud computing and the business implications, it is very important to have such architecture in place. I proposed a fuzzy-based model for cloud computing.

The following sub-sections of this chapter summarize the thesis contributions to the state of the art, and take into consideration the research issues stated above.

9.2.1 Development of a Dynamic Pricing Scheme for Cloud Services

In this thesis, a dynamic pricing scheme is introduced for cloud service customers and cloud service providers. In the proposed scheme, a cloud market agent is used to provide the matching process and negotiation about service level objectives for cloud users. In this scheme, pricing functions are proposed to control the cost level from the perspective of cloud customers, and to control the resource allocation and maximizing the revenue from the perspective of cloud service providers. I discussed the problem of how to announce the service price when the demand of service and the resources are not stable in the cloud computing market. I use weighting parameters to control the increasing and decreasing amount of pricing when the cloud market is in high and low states. The conducted simulations show the proposed dynamic pricing scheme for cloud computing provides scientific results of improving the revenue for cloud service providers and help cloud customers to rent a high quality of cloud services.

9.2.2 Development of a Methodology or SLA of Cloud Platforms

A service level agreement is a document that includes a description of the agreed service, service level parameters, guarantees, and actions for all cases of violation. The SLA is very important as a contract between consumer and provider. The main idea of SLAs is to give a clear definition of the formal agreements about service terms such as performance, availability and billing. It is important that the SLA include the obligations and the actions that will be taken in the event of any violation, with clearly expressed and shared semantics between each party involved in the online contract.

From the discussion of SLA problems in the context of cloud computing, the following issues emerge:

1. The existing frameworks focus more on the technical attributes than on the security and management aspects of services.

2. The proposed structures of SLAs in the above domains do not include a clear definition of the relationship between levels of violation and the cost of services.

3. Most of the above studies do not integrate a framework of trust management of the service provider with the collected data from monitoring systems of SLAs.

4. The concepts and definitions of service objectives and service descriptions included in SLAs are not easy to understand, especially for business decision-makers.

5. The proposed works for cloud environments focus more on the evaluation of virtualization machines on local servers than on existing cloud service providers.

6. Most of the proposed structures of SLAs are defined by technical experts.

To solve these problems, I propose a methodology for service level agreements for the users of cloud computing (Chapter 6). I consider the main types of cloud services (IaaS, PaaS, DaaS and SaaS). For each type of cloud service, I investigate the main metrics that can be used to obtain a reliable method for negotiating and signing the SLA in the context of cloud computing.

9.2.3 Development a Methodology for Performance and Measurement for Cloud Services

The different architecture of cloud services provided by IaaS should be known before running applications on any IaaS platform. Unfortunately, here it is not appropriate to comment on the processor type to be used; however, users could consider the percentage of different processors being used. This would help them predict the level of cloud service performance and also the ability to conduct experiments in different scenarios. Also, users should consider using equivalent types of virtual machines while comparing various applications using IaaS platforms.

I also considered the effect of availability in different zones of IaaS and concluded that it is very important that users select one zone of availability and not leave it to the scheduler for the application quality. It was also noted that due to the differences in performance guarantee of service level agreements, companies are not able to use IaaS for performance stability-based applications. Thus, cloud should offer performance-based service level agreements to the users. The difference between the use of Windows-based VMs and Linux-based VMs was also noticed; hence, users should be enabled to choose virtual machines with particular hardware settings in Amazon EC2 such as CPU, memory, network etc.

Cloud computing is a new form of technology whose infrastructure, developing platform, software, and storage can be delivered as a service in a pay-as-you-use cost model. Intelligent use of resources in cloud computing may help cloud customers to reduce the large amount of IT investments as well as operational costs. However, for critical business applications and more sensitive information, cloud providers must be selected based on high levels of performance and trustworthiness. In order to use cloud services, it is very important to understand the performance of the cloud infrastructure provided by clouds. In chapter 7, I evaluate the EC2 instances as an example to examine the stability of most types of VMs which are provided by Amazon. I demonstrate that the large types of VMs have the best stability in terms of performance. So, as a service level agreement, performance metrics can be used as a good parameter in the agreement. But for small types of VMs, it is important to improve the stability of performance before signing any agreement between cloud provider and user.

9.2.4 Development a Trust and Reputation Model for Cloud Services

Cloud computing has emerged as an effective technology where the computing infrastructure, networking routers, software, and developing platform are delivered as services available for users at any time and where they can access the Internet [1]. With the increase in public cloud providers, cloud consumers face various challenges such as the security, privacy, and discovery of reliable resource providers. One of the challenges presenting the greatest barrier to the adoption of external cloud providers is the issue of whether cloud users can trust cloud providers to deliver what they promise. Different trust and reputation models have been proposed in the literature of information technology. But none of these models is discussed in the context of cloud computing. In this thesis, I propose a trust model using the fuzzy logic approach with the first-hand experiences trust values in order to determine a reliable method to select the most secure providers of cloud resources.

9.3 FUTURE WORK AND RESEARCH DIRECTIONS

As discussed throughout this thesis, there are many further research directions that could be pursued based on my contributions. Dynamic pricing models for multi-domain cloud communities and the large market of cloud services have still not been developed for practical use. Also, complicated issues related to service level agreements and performance benchmarking for cloud computing still need to be provided in more reliable and flexible systems. I intend to continue this research by investigating a number of challenges related to the problems addressed in this thesis. This section discusses some issues that will direct my research approach in the future.

9.3.1 Improve the Allocation Approach Based on Dynamic Pricing Model

This research has presented a methodology for pricing cloud services in dynamic time, with a dynamic demand feature and value adding to my understanding of the price approaches for services of cloud computing. Based on the preceding analysis and discussion in Chapter 5, it appears that the market for cloud computing services is still evolving and certain challenges still lie ahead. For example:

1. Cloud users currently do not have any control of the hardware and software management and thus are limited in responding to emergency situations.

2. The actual ownership of the data is unknown and thus security aspects of the information are compromised, thereby creating fears among the customers. One way to address this is to develop robust security solutions for cloud computing services by incorporating appropriate properties in cloud services' service level agreements.

3. Many researchers argue that the practice of changing providers is difficult and the high cost of one provider means a locked-in consequence for the cloud service customer.

4. As the cloud computing market is still in its early stages, there is no implementable common price structure/mechanism or pricing schemes. Commonly used approaches are resources pricing, pay-as-you-go and payment models, and they are utilized on all

layers (i.e. IaaS, PaaS and SaaS). There is a need to study consumer preferences on other layers besides the infrastructure layer.

5. In addition to the price models discussed in this thesis, other models such as the fixed pricing approach have also been proposed in [2]. It calculates the bid price based on an additive method. The downside of this approach is its addition to the bid price in each time segment leading to a lower degree of flexibility in the additive function. Furthermore, with the fixed approach the time segment is restricted, thus making it applicable in cases where usage starts at the end of the time segment. There is a need to extend the use of the fixed approach under a continuous usage regime.

9.3.2 Implementation of SLAs Framework for Cloud Computing

The existing cloud platforms in the market of cloud services do not offer a clear guarantee or well-defined service level agreements (SLAs) to satisfy different interests of cloud customers who need to ensure the continuity if their business for the long term. Cloud users do not want just a certain percentage of guarantees of availability; they require a more specific definition of an SLA that can be used as an agreement about the required quality or level of services for different applications. Based on the model of cloud service delivery, various parameters of SLA metrics can be included in the structure of SLA for cloud services. On the other hand, users of PaaS, SaaS, and DaaS may ask for specialized types of SLA parameters that should be included in SLA agreements. As a future work, I intend to investigate further issues regarding SLA in cloud computing especially with regards to the correlation between business objectives of cloud providers and the objectives of performance. Also, I will develop a methodology to implement real environment of experiments to demonstrate the importance of using SLA in each type of cloud service.

9.3.3 Investigating Further Issues for Performance Measurements in Cloud Computing

Real-world-based applications experiments of the performance differences of the most widely used cloud are conducted on Amazon EC2. It was illustrated that there is a difference in the performance of cloud applications. The performance has been benchmarked frequently over the period of the experiments. The analysis distinctly shows that there is a significant difference between small and large types of virtual machines hosted on Amazon EC2. The standard deviations for CPU, data transferring and network bandwidth which were observed

in my experiments show large differences between the performances of evaluated VMs on Amazon EC2 and a local machine. One of the reasons for variation is the types of resources used in both scenarios. The difference in results on the cloud was compared against a local machine which confirmed that the micro benchmark had a significantly higher performance difference than did the Amazon EC2 VMs. Since runtime is also affected, experiments should be conducted on an Amazon EC2 platform with consideration given to external effects. It was also observed that as the difference is high and the distribution of measurements is not normal, it would not be an easy task to determine the intervals for measurements. However in order to determine which system is better, EC2 may be used as described earlier and in [3].

Many interesting opportunities for future research have also arisen. First, the results could be discussed with IaaS providers and better ways to introduce service level agreements could be devised. Also, the way cloud providers could provide virtual machines is important so that meaningful experiments are conducted. It would also be interesting to ascertain whether other cloud providers experience the problem which is considered in this part of my thesis. All this has been left for future work for which new technology and algorithms would be required and which is beyond the scope of this study.

9.3.4 Investigating Difference Approaches for Trust Calculation in Cloud Computing

In terms of trust, the challenge for cloud architecture is to implement cloud computing systems in such a way as to increase the trust level of cloud users and fulfil customer requirements. Users of cloud services may take legal action in the case of any violations of service level agreements. Also, cloud providers may face legal action and damage the reputation of their organizations if they violate the terms of the SLA.

In the case of cloud architecture, the dynamic nature of the allocation of cloud resources raises the need to develop novel frameworks that should be implemented to minimize the calculation time and provide more reliable results. These frameworks should take into account the QoS aware based systems and integrate QoS evaluation methods with the trust solution for cloud computing. In my proposed solution of trust evaluation for cloud services, I use a fuzzy logic approach. My proposed future work is to continue to evaluate the proposed solution in the context of measuring the calculation time of trust value. Also, I will compare the results with different method of calculation of trust value in other domains such as web services and wireless service domains.

9.3.5 Extending the Proposed Model of Dynamic Pricing of Cloud Services

In Chapter 5, I investigated the issues related to how to correlate the price of cloud services from both sides of cloud customers and cloud providers. I focused on the notion of dynamic demand of cloud services and its impacts on the price of services. More challenges should be considered to implement more attractive solution for pricing cloud services. As a future work, I will focus on how to use the dynamic notion of resource allocation and dynamic demand of cloud services to develop a flexible model for the price of cloud services. I can use the same technique that is already being used in Smart Grid Technology [5]. In Smart Grid Systems, residents can generate and sell power at different times at different prices. The price depends on the size of the power demand. If there are large numbers of users who want to buy power, residents who generate power can propose a higher price for the generated power. However, in times of low demand for power such as warmer months, residents who generate power can propose a low price for generated power.

In cloud computing, there are many complicated issues regarding the control of resources allocation of cloud services. In future work, I will continue to investigate more factors that may be considered to have a high impact on the price of cloud services. Then, I will develop a more flexible model to be used by cloud customers. The proposed model will help cloud users to select the best time and best providers of cloud services.

9.3.6 Developing Monitoring System for Cloud Services

In a cloud environment, there are many criteria that should be evaluated, monitored and used with trust solutions. I presented a methodology to evaluate the performance of resources in cloud computing. The evaluation criteria such as response time, memory throughput, and network bandwidth may provide required information for specific users of cloud computing. But, for many cloud customers, these criteria are not enough to build a relationship that has a high level of trust. So, there is a need to develop a comprehensive system to monitor the important metrics of cloud resources. As future work subsequent to this thesis, I will define more criteria for the four types of cloud services and I will develop a reliable model for monitoring the resources metrics of cloud computing. To validate the proposed model, I will develop a prototype system to be integrated with the most-used type of virtualization environment.

9.3.7 Extending my Proposed Fuzzy-Based Trust Model

In Chapter 8, I proposed a Takagi-Sugeno fuzzy-inference approach to evaluate the trustworthiness of one type of cloud service which is Infrastructure as a Service (IaaS). In future work, I will extend my proposed model of trust to consider other types of cloud services including Platform as a Service (PaaS), Software as a Service (SaaS), and Database as a Service (DaaS). To apply the Takagi-Sugeno approach to these types of services, I will develop a cloud computing expert-based survey. The results of this survey will be analysed to train the proposed model. I will focus on the experts and normal users of cloud computing in order to obtain a more accurate fuzzy system for each type of cloud service. A general trust model can be developed after creating a fuzzy-based trust model for all types of cloud services. The use of a general trust model for cloud computing will help cloud users to use a customized common trust system based on their preferences and the types of cloud services they require.

REFERENCES

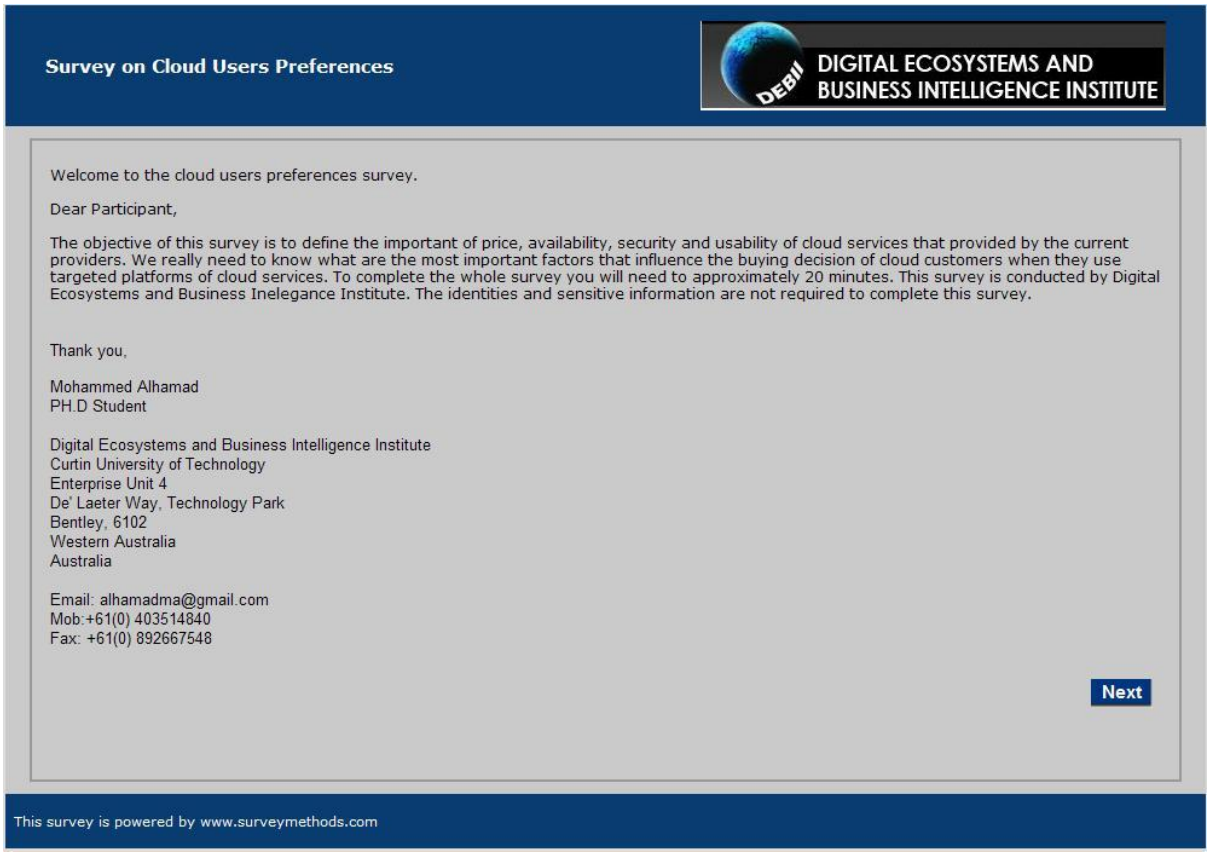
[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, and I. Stoica, *"Above the clouds: A berkeley view of cloud computing"*,EECS Department, University of California, Berkeley, Tech Rep, UCB/EECS-2009-28, 2009.

[2] J. Joutsensalo, T. Hämäläinen, K. Luostarinen, and J. Siltanen, *"Adaptive scheduling method for maximizing revenue in flat pricing scenario*", AEU-International Journal of Electronics and Communications, vol. 60, pp. 159-167, 2006.

[3] A. Mohammed, D. Tharam, W. Chen, and C. Elizabeth, *"Response Time for Cloud Computing Providers"*, Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services (iiWAS '10), ACM, pp. 603-606, 2010.

[5] Omid Siakani, Omar Hussian, Tharam Dillon,Azadeh Tabesh, *"Intelligent Decision Support System for Including Consumers' Preferences in Residential Energy Consumption in Smart Grid",* Second InternationalConference on Computational Intelligence, Modelling andSimulation, 2010.

APPENDIX A SURVEY OF CLOUD USERS PREFERENCES



Survey on Cloud Users Preferences

DIGITAL ECOSYSTEMS AND BUSINESS INTELLIGENCE INSTITUTE

Welcome to the cloud users preferences survey.

Dear Participant,

The objective of this survey is to define the important of price, availability, security and usability of cloud services that provided by the current providers. We really need to know what are the most important factors that influence the buying decision of cloud customers when they use targeted platforms of cloud services. To complete the whole survey you will need to approximately 20 minutes. This survey is conducted by Digital Ecosystems and Business Inelegance Institute. The identities and sensitive information are not required to complete this survey.

Thank you,

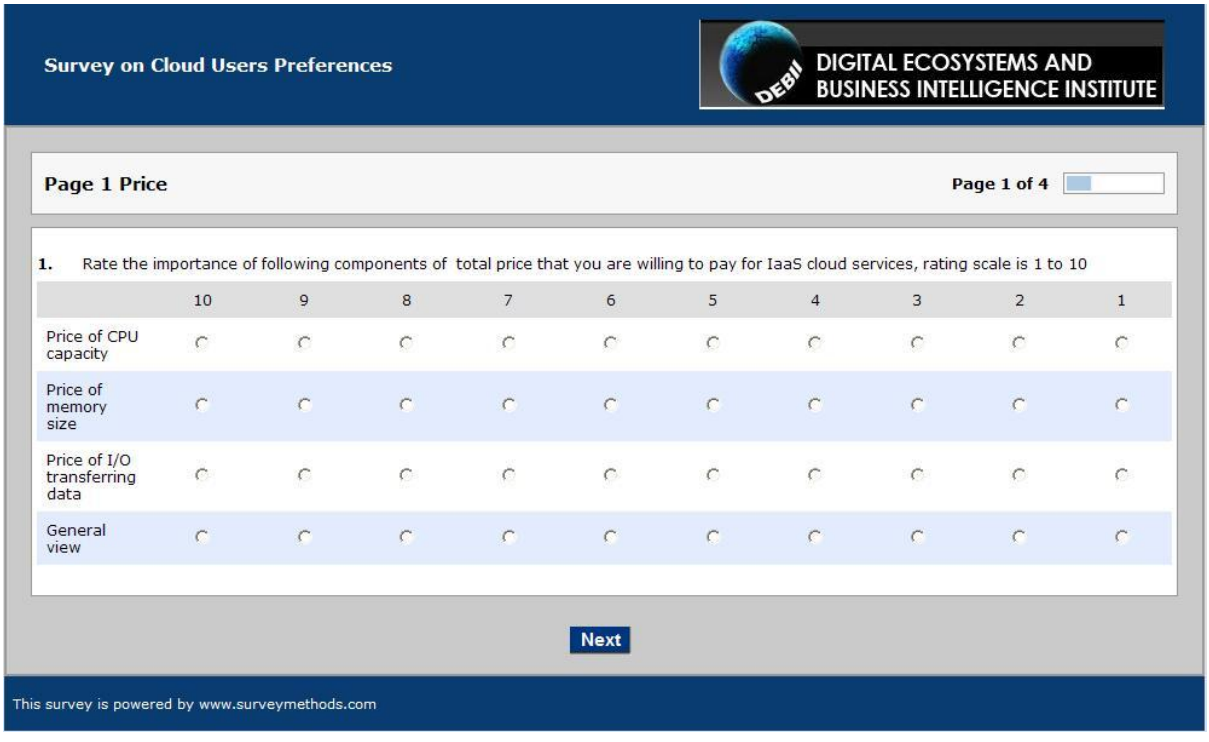
Mohammed Alhamad
PH.D Student

Digital Ecosystems and Business Intelligence Institute
Curtin University of Technology
Enterprise Unit 4
De' Laeter Way, Technology Park
Bentley, 6102
Western Australia
Australia

Email: alhamadma@gmail.com
Mob:+61(0) 403514840
Fax: +61(0) 892667548

Next

This survey is powered by www.surveymethods.com



Survey on Cloud Users Preferences

DIGITAL ECOSYSTEMS AND BUSINESS INTELLIGENCE INSTITUTE

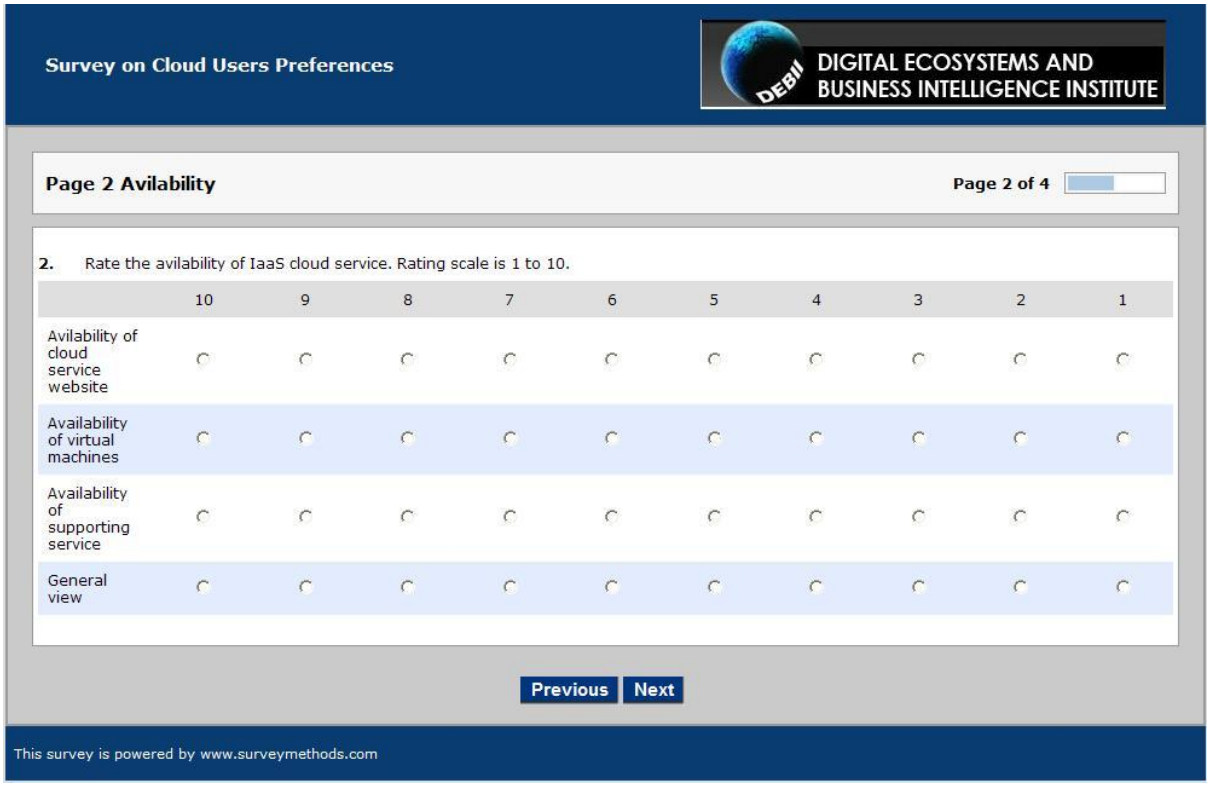
Page 1 Price Page 1 of 4

1. Rate the importance of following components of total price that you are willing to pay for IaaS cloud services, rating scale is 1 to 10

	10	9	8	7	6	5	4	3	2	1
Price of CPU capacity	○	○	○	○	○	○	○	○	○	○
Price of memory size	○	○	○	○	○	○	○	○	○	○
Price of I/O transferring data	○	○	○	○	○	○	○	○	○	○
General view	○	○	○	○	○	○	○	○	○	○

Next

This survey is powered by www.surveymethods.com



Survey on Cloud Users Preferences

DIGITAL ECOSYSTEMS AND BUSINESS INTELLIGENCE INSTITUTE

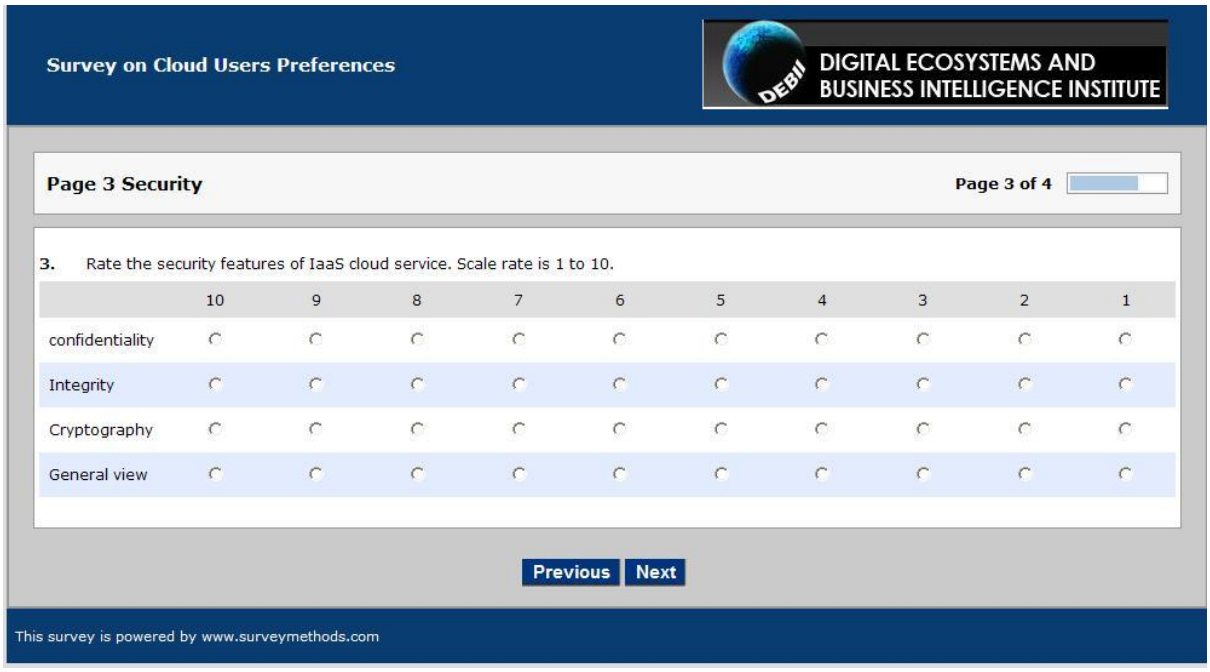
Page 2 Avilability Page 2 of 4

2. Rate the avilability of IaaS cloud service. Rating scale is 1 to 10.

	10	9	8	7	6	5	4	3	2	1
Avilability of cloud service website	○	○	○	○	○	○	○	○	○	○
Availability of virtual machines	○	○	○	○	○	○	○	○	○	○
Availability of supporting service	○	○	○	○	○	○	○	○	○	○
General view	○	○	○	○	○	○	○	○	○	○

Previous Next

This survey is powered by www.surveymethods.com



Survey on Cloud Users Preferences

DIGITAL ECOSYSTEMS AND BUSINESS INTELLIGENCE INSTITUTE

Page 3 Security Page 3 of 4

3. Rate the security features of IaaS cloud service. Scale rate is 1 to 10.

	10	9	8	7	6	5	4	3	2	1
confidentiality	○	○	○	○	○	○	○	○	○	○
Integrity	○	○	○	○	○	○	○	○	○	○
Cryptography	○	○	○	○	○	○	○	○	○	○
General view	○	○	○	○	○	○	○	○	○	○

Previous Next

This survey is powered by www.surveymethods.com

Survey on Cloud Users Preferences

DIGITAL ECOSYSTEMS AND BUSINESS INTELLIGENCE INSTITUTE

Page 4 Usability Page 4 of 4

4. Rate the usability of IaaS cloud service. Rating scale is 1 to 10.

	10	9	8	7	6	5	4	3	2	1
System tutorials	○	○	○	○	○	○	○	○	○	○
System feedback	○	○	○	○	○	○	○	○	○	○
Error recovery	○	○	○	○	○	○	○	○	○	○
General view	○	○	○	○	○	○	○	○	○	○

Previous Submit

This survey is powered by www.surveymethods.com

SELECTED PUBLICATIONS

Response Time for Cloud Computing Providers

Mohammed Alhamad, Tharam Dillon, Chen Wu, Elizabeth Chang
Digital Ecosystems and Business Intelligence Institute (DEBII)
Curtin University of Technology
Perth, Australia
Mohammed.Alhamad@postgrad.curtin.edu.au,
Tharam.Dillon@cbs.curtin.edu.au, Chen.Wu@cbs.curtin.edu.au
Elizabeth.Chang@cbs.curtin.edu.au

ABSTRACT

Cloud services are becoming popular in terms of distributed technology because they allow cloud users to rent well-specified resources of computing, network, and storage infrastructure. Users pay for their use of services without needing to spend massive amounts for integration, maintenance, or management of the IT infrastructure. This creates the need for a reliable measurement methodology of the scalability for this type of new paradigm of services. In this paper, we develop performance metrics to measure and compare the scalability of the resources of virtualization on the cloud data centres. First, we discuss the need for a reliable method to compare the performance of cloud services among a number of various services being offered. Second, we develop a different type of metrics and propose a suitable methodology to measure the scalability using these types of metrics. We focus on the visualization resources such as CPU, storage disk, and network infrastructure. Finally, we compare well-known cloud providers using the proposed approach and conclude the recommendations. This type of research will help cloud consumers, before signing any official contract to use the desired services, to ascertain the ability and capacity of the cloud providers to deliver a particular service.

Keywords

Performance, SLA, Cloud computing, Trust management

1. INTRODUCTION

There have been various definitions proposed in the literature of cloud computing [1-3].

iiWAS2010, 8-10 November, 2010, Paris, France.

In this paper, we adopted and considered the definition provided by U.S. NIST (National Institute of Standards and Technology) that describes cloud computing as "... a model for enabling convenient, on demand network access to a share pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management afford or service provider interaction" [1]. In other words, as shown in Figure 1, cloud computing is a framework by means of which virtualized infrastructure resources are delivered as a service to customers by using a public network which is the Internet [4-6].

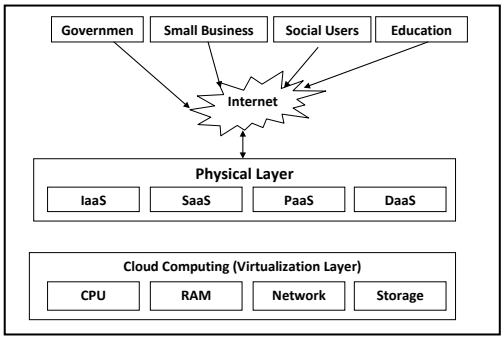


Figure 1. Cloud computing architecture

The cloud customers can range from big organizations, small business and developers to individual users. In this paper, we will refer to such customers as 'users'. One of the advantages of having such a framework is that users do not need to buy costly physical infrastructure or software, but they can use them over a virtual environment from other users at a much lower price, thereby reducing their operational and maintenance costs. For example, Salesforce.com developed a customer relationship management solution (CRM) and delivered this as a cloud service not as a package of software. Salesforce.com customers can use this type of service using a basic machine with an Internet browser [7]. There are four main delivery models of cloud services with such a paradigm. They are:

1. Infrastructure as a service (IaaS): In such architectures, users can use the visualization resources as a fundamental infrastructure for their applications. These resources may be a CPU, network, or storage. Cloud users can manage the resources and assign rules for end users [8].
2. Database as a service (DaaS): Such architectures allow users to rent a specific size of storage for a specific period of time. Users are not required to manage the integration

or the scaling of the infrastructure. Database providers take the responsibility for integration, privacy, and security of users' data [9].

3. Platform as a service (PaaS): In such architectures, users use all facilities on the cloud to develop and deliver their web application and services to the end users. PaaS services may include development, integration, testing or the storage resources to complete the life cycle of services [10].
4. Software as a service (SaaS): In such architectures, users connect with the service providers to use the application, but they do not control the infrastructure, operating system or network infrastructure [10, 11].

Each of these delivery models is above the required hardware and virtualization model as shown in Figure 2.

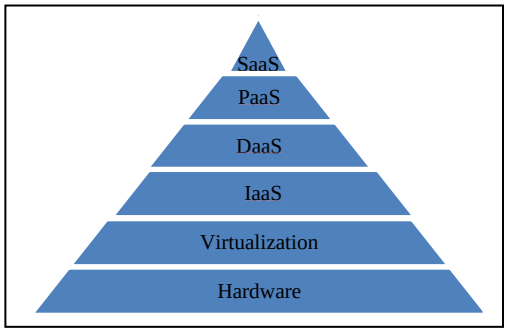


Figure 2. Cloud computing stack layers

No matter what type of delivery model is being used, there are five essential factors or characteristics that have to be satisfied to achieve smooth computing in a cloud computing environment. They are:

a) On-demand self-service: On demand self-service refers to the availability of the required resources (such as CPU power, network etc) as and when the user needs it. Further, this should be without any human intervention [12].

b) Broad network access: As the interacting medium between the different users is the Internet, there should be a broad network access available that allows for the seamless interaction of different applications across different heterogeneous platforms [13].

c) Resource Pooling: A cloud provider should support multi-tenancy of its resources for maximising the efficiency of its infrastructure. For example, it should be able to dynamically assign the required resources to the consumer according to its demand [14].

d) Rapid Elasticity: It should be flexible according to the computing resources required for the customers. For example, there is no up-front commitment and the customers should be able to release the resources once their work is done [15].

e) Measure of Service: There should be a framework that measures the usage of each user according to the resources that are being used by it [16].

To test the performance of hardware or real applications, test and evaluations rules should be defined and implemented to serve as a comparative tool for performance metrics. Instead of having a large investment and lengthy time to use non-reliable providers, benchmarks [17] assist decision makers to save money and choose the cloud providers who fulfil their objectives. Based on the study objectives or research, an appropriate benchmark can be chosen and a targeted application or system is deployed. Then, the results can be analysed using different techniques to obtain the final recommendation. In this paper, we test the stability of performance of different types of Amazon EC2 instances in order to investigate the use of a performance parameter as the main criterion for service level agreements (SLA) between cloud providers and their customers. Before deploying our application on cloud instances, the same application was executed on a local machine and the response time in this experiment was more stable. So, the standard deviation was almost 1.01. But in the cloud environment, the results vary based on the type of EC2 instances. More details about experiment results are discussed in the experiments section.

In our study, we ran a series of experiments on Amazon EC2 cloud over a different number of times. For each time period, we evaluated the response time of five types of Amazon EC2 instances. The main contribution of our study is testing the isolation across the same hardware of virtual machines which are hosted by a cloud provider. There are different ways to evaluate the scalability of cloud providers, for instance, evaluating of throughput of network, disk performance, and capacity of RAM. In this paper, we use the CPU performance as a main parameter for cloud performance, and we measure the execution time of the deployed application over five types of Amazon EC2 instances. We recorded the response time every two hours during several days of experimentation.

The rest of this paper is structured as follows. We discuss related work in Section 2. The methodology of our contribution is presented in Section 3. We present the results and our evaluation in Section 4 and conclude in Section 5.

2. LITERATURE REVIEW

Several studies on the scalability of virtual machines already exist. Most of these studies considered the measurement of performance metrics on the local machines. The background loads of tested machines are controlled to compare the results of performance with a different scale of loads. To the best of our knowledge, to date, no such methodology has been developed to study the performance for cloud providers by considering the use of different metrics of performance. For example, Evangelinos and Hill [18] evaluated the performance of Amazon EC2 to host High Performance Computing (HPC). They use 32-bit architecture for only two types of Amazon instances. In our study, we run various experiments on most types of Amazon EC2 instances. These instances are: small, large, extra large, high CPU, medium, and high CPU extra large instance. Jureta, and Herssens [19] propose a model called QVDP. This model has three functions: specifying the quality level, determining the dependency value, and ranking the quality priority. These functions consider the quality of services from the customers' perspective. However, the performance issues related to cloud resources are not discussed and details are missing regarding the correlation of the quality model with the costing model of services. Cherkasova and Gardner [20] use a performance benchmark to analyse the scalability of disk storage and CPU capacity with Xen Virtual Machine Monitors. They measure the performance parameters of visualization infrastructure that are already deployed in most data centres. But they do not measure the scalability of cloud providers using the visualization resources. However, our proposed work profiles the performance of virtualization resources that are already running on the infrastructure of cloud providers such Amazon EC2 services.

3. METHODOLOGY

3.1 Benchmark

We ran Java application on Amazon EC2 over a period of days. We used our benchmark to measure the variations in the performance of CPU for the five types of Amazon EC2 instances. If the collected results show that the execution time of chosen application is stable, then this will provide evidence that a cloud infrastructure is able to run applications which need stability of response time. If the collected results have sizeable variations in response time, then the particular cloud provider is not able to host applications that consider the response time as one of main objectives in the service level agreement (SLA).

3.2 Experiment Setup

We used different types of virtual machines in terms of CPU capacity, RAM size, and bandwidth of disk and network. Table 1 show the features of Amazon EC2 instances that were used in our experiments.

Table 1. Features of Amazon EC2 instances

Instance Type	EC unit	Cores	Architecture	Disk (GB)	RAM (GB)
Small	1	1	32	160	1.7
Medium (H-CPU)	5	2	32	350	1.7
Large	4	2	64	850	7.5
Extra Large	8	4	64	1690	15
Extra Large (H-CPU)	20	8	64	1690	7

There are different uses of cloud computing technology and the results of the performance using different applications are different. The performance comparison is not fair in this case. So, we deploy one Java application on all types of cloud instances and we collect results without changing the scalability of our application. Our goal is to see how the usage changes when the backload is changed in the same machine in the cloud data centre. The proposed metrics to measure the scalability of cloud providers will evaluate throughput of network, disk performance, and capacity of RAM. In this paper, we use the CPU performance as a main parameter for cloud performance; in our future work, we will use the other metrics and evaluate the same types of Amazon EC2 that were used in this paper.

4. EXPERIMENT RESULTS

In this section, we compare the response time of selected VMs which are provided by EC2. The performance metric we are measuring does not include the booting and installing time which has various measurements between 80 and 220 seconds. Also, the response time reported does not include the transferring of input and output data. Table 2 shows the 5 samples of performance metrics of EC2 instances.

Table 2. Samples of response time of Amazon EC2 instances

Small	Medium (H-CPU)	Large	Extra Large	Extra Large (H-CPU)	Local Machine
656	375	110	125	125	360
734	375	125	172	125	360
844	375	109	124	125	359
650	438	172	187	125	360
STDDEV 122.9	STDDEV 23.5	STDDEV 48.9	STDDEV 27.7	STDDEV 7.2	STDDEV 1.1
Average 769.3	Average 383.2	Average 126.9	Average 153.8	Average 129.6	Average 359.8

In the performance metrics, the best stability was for the Extra large (H-CPU) type. This is due to the fact that Extra large (H-CPU) has the best resources of CPU power. Figure 3-7 show the stability of the performance on the selected types of VMs.

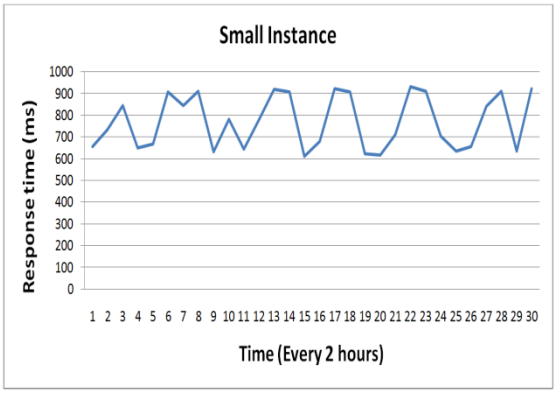


Figure 3. Response time of small instances

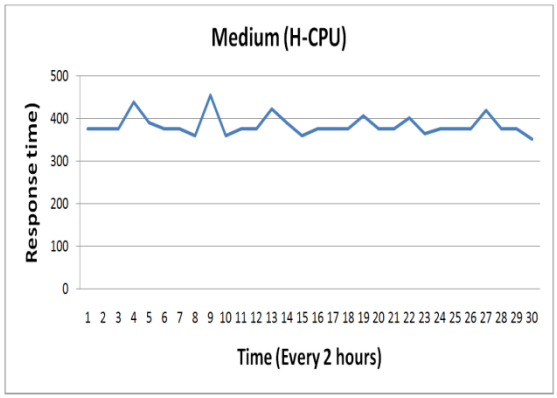


Figure 4. Response time of medium instances

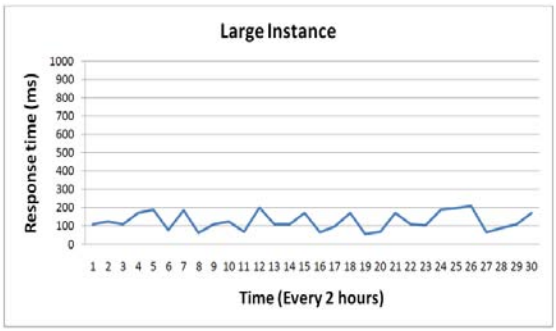


Figure 5. Response time of large instances

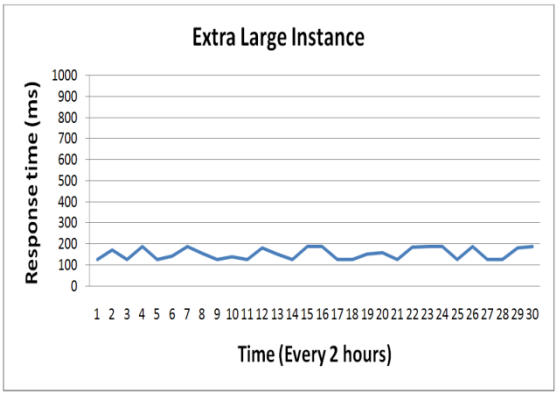


Figure 6. Response time of extra large instances

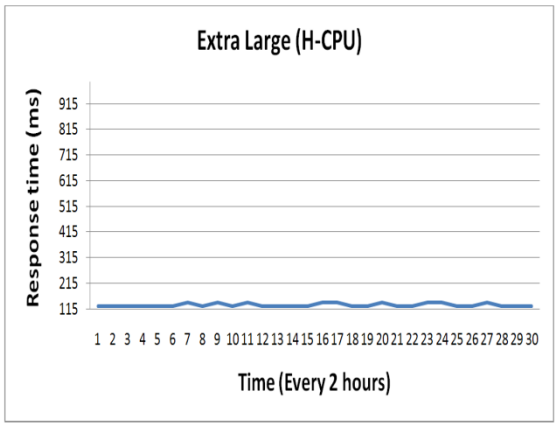


Figure 7. Response time of extra large (H-CPU) instances

5. CONCLUSION

Cloud computing is a new form of technology, whose infrastructure, developing platform, software, and storage can be delivered as a service in a pay-as-you-use cost model. Intelligent usage of resources in cloud computing may help cloud customers to reduce the large amount of IT investments as well as operational costs. However, for critical business application and more sensitive information, cloud providers must be selected based on high level of performance and trustworthiness. To use cloud services, it is very important to understand the performance of the cloud infrastructure provided by clouds. In this paper, we evaluate the EC2 instances as example to examine the stability of most types of VMs which provided by Amazon. We demonstrate that the performance of Extra large high CPU has the best stability of performance. So, as a service level agreement, response time can be used as a good parameter in the agreement. But for small, large, and extra large instances, it is important to improve the stability of response time before signing any agreement between cloud provider and user.

6. REFERENCES

[1] P. Mell and T. Grance, Draft nist working definition of cloud computing, 2009.

[2] J. Napper and P. Bientinesi, Can cloud computing reach the top500?, 2009, pp. 17-20.

[3] Y. Chen, et al., What's New About Cloud Computing Security?, 2010.

[4] R. Buyya, Market-Oriented Cloud Computing: Vision, Hype, and Reality of Delivering Computing as the 5th Utility, 2009, p. 1.

[5] A. Marinos and G. Briscoe, Community cloud computing, CoRR, abs/0907.2485, 2009.

[6] P. T. Jaeger, et al., Cloud computing and information policy: Computing in a policy cloud?, Journal of Information Technology & Politics, vol. 5, pp. 269-283, 2008.

[7] M. Nelson, Building an Open Cloud, Science, vol. 324, p. 1656, 2009.

[8] D. Hilley, Cloud Computing: A Taxonomy of Platform and Infrastructure-level Offerings, 2009.

[9] H. Cai, et al., Customer Centric Cloud Service Model and a Case Study on Commerce as a Service, 2009, pp. 57-64.

[10] D. Cerbelaud, et al., Opening the clouds: qualitative overview of the state-of-the-art open source VM-based cloud management platforms, 2009, pp. 1-8.

[11] J. Muller, et al., Customizing Enterprise Software as a Service Applications: Back-End Extension in a Multi-tenancy Environment, 2009, p. 66.

[12] B. Sotomayor, et al., Virtual infrastructure management in private and hybrid clouds, IEEE Internet Computing, vol. 13, pp. 14-22, 2009.

[13] D. Nurmi, et al., The eucalyptus open-source cloud-computing system, 2009, pp. 124-131.

[14] M. Zeller, et al., Open standards and cloud computing: Kdd-2009 panel report, 2009, pp. 11-18.

[15] T. Dillon, et al., Cloud Computing: Issues and Challenges, 2010, pp. 27-33.

[16] J. Nunamaker Jr, et al., Systems development in information systems research, Journal of Management Information Systems, pp. 89-106, 1990.

[17] P. Donohoe, A Survey of Real-Time Performance Benchmarks for the Ada Programming Language, ed: Citeseer, 1987.

[18] C. Evangelinos and C. Hill, Cloud Computing for parallel Scientific HPC Applications: Feasibility of running Coupled Atmosphere-Ocean Climate Models on Amazon's EC2, ratio, vol. 2, p. 2.34, 2008.

[19] I. Jureta, et al., A comprehensive quality model for service-oriented systems, Software Quality Journal, vol. 17, pp. 65-98, 2009.

[20] L. Cherkasova and R. Gardner, Measuring CPU overhead for I/O processing in the Xen virtual machine monitor, 2005, p. 24.

Conceptual SLA Framework for Cloud Computing

Mohammed Alhamad, Tharam Dillon, Elizabeth Chang
Digital Ecosystems and Business Intelligence Institute (DEBII)
Curtin University of Technology
Perth, Australia
Mohammed.Alhamad@postgrad.curtin.edu.au,
Tharam.Dillon@cbs.curtin.edu.au, Elizabeth.Chang@cbs.curtin.edu.au

Abstract. **Cloud computing has been a hot topic in the research community since 2007. In cloud computing, the online services are conducted to be pay-as-you-use. Service customers need not be in a long term contract with service providers. Service level agreements (SLAs) are agreements signed between a service provider and another party such as a service consumer, broker agent, or monitoring agent. Because cloud computing is a recent technology providing many services for critical business applications, reliable and flexible mechanisms to manage online contracts are very important. This paper presents the main criteria which should be considered at the stage of designing the SLA in cloud computing. Also, we investigate the negotiation strategies between cloud provider and cloud consumer and propose our method to maintain the trust and reliability between each of the parties involved in the negotiation process.**

Index Terms: SLA, Negotiation, Cloud computing, Trust management

I. Introduction

Cloud computing has been a hot topic in the research community recently. In cloud computing, the online services are conducted on a pay-as-you-use basis. It is not necessary to be in a long term contract with service providers [1]. In this case, cloud customers can save large amounts of budget spent on operating, managing and transferring services. Cloud computing can be described as a new form of IT environment which provides dynamic, flexible and scalable virtualization of resources. Examples of current cloud providers include: Amazon EC2 [2] (infrastructure cloud provider), Azure [3] from Microsoft (platform cloud provider), and for an application cloud provider, there is Google Docs [4] . In cloud computing, virtualization technology is built on top of the infrastructure in order to optimize the use of resources and provide flexible solutions for users. An important element that provides some degree of assurance to both users and providers of these cloud resources is the Service Level Agreements which define the scope of usage and provision of resources. Cloud consumers need an SLA before they transfer their infrastructure to cloud data centres, to provide certainty regarding the resources provided and the ability to reach the desired level of productivity. Cloud providers need an SLA to define the trust and quality of services they provide to users as well as an agreed framework for costs and charges. The research on SLA and QoS metrics has been considered by many researchers in business and service-oriented architecture such as e-commerce and web services. However, SLA metrics in these technologies are not suitable for cloud computing as the nature and type of resources being provided and delivered is different. So, new SLA models are still required to provide flexible method for negotiation and the signing of electronic contracts between consumers and providers. The main contributions of this paper are summarized as follows:

1) Investigating and analysing the main requirements to establish an effective model for SLA in cloud computing
2) Defining dynamic SLA metrics for different groups of cloud users

The remainder of this paper is structured as follows: Section 2 defines SLAs and describes the main characteristics of SLAs in cloud computing. The existing standards for SLA contracts are presented in Section 3. In Section 4, properties and main criteria for SLA in cloud computing are described. Also, in this section, the negotiation model and negotiation scenarios for cloud computing are discussed. Section 5 concludes the paper.

II. Characteristics of Service Level Agreement

A. Definition

A Service level agreement is a document that includes a description of the agreed service, service level parameters, guarantees, and actions and remedies for all cases of violations [5]. The SLA is very important as a contract between consumer and provider. The main idea of SLAs is to give a clear definition of the formal agreements about service terms like performance, availability and billing. It is important that the SLA include the obligations and the actions that will

be taken in the event of any violation, with clearly shared semantics between each party involved in the online contract.

B. *Properties of SLAs*

The SLA is a legal format documenting the way that services will be delivered as well as providing a framework for service charges. Service providers use this foundation to optimize their use of infrastructure to meet signed terms of services. Service consumers use the SLA to ensure the level of quality of service they need and to maintain acceptable business models for long term provision of services. The following are the main requirements of the SLA:

- SLA format should clearly describe a service so that the service consumer can easily understand the operation of the services
- Present the level of performance of service
- Define ways by which the service parameters can be monitored and the format of monitoring reports
- Penalties when service requirements are not met
- Present the business metrics such as billing and stipulate when this service can be terminated without any penalties being incurred

This is the requirements for SLAs in the general environment of services. Later, we present the main requirements which the SLA should implement in order to integrate with the cloud computing architecture.

C. *Functional and non-functional requirements for cloud users*

Functional requirements and non-functional requirements of cloud services should be met to fulfil the need of consumers. In this section, classification of cloud computing requirements from the perspective of the cloud consumer is presented, helping to provide a good understanding of the proposed framework in Section 4. For each type of cloud service, there are different requirements. Figure 1 shows the categorization of cloud computing services and requirements for each service.

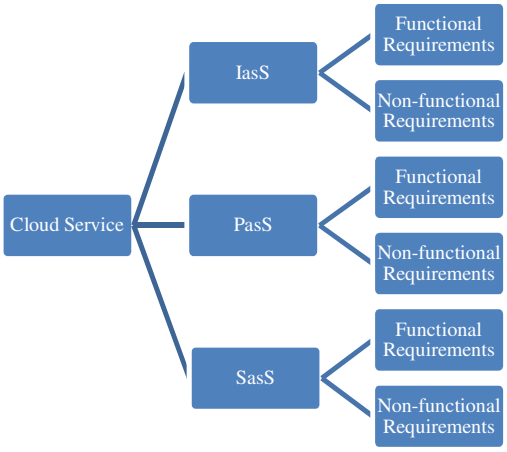


Figure 1. Categorization of requirements for cloud services

In this paper, we focus on the non-functional requirements of services such as availability, scalability and response time. Based on the more important non-functional requirements, we define the SLA parameters for each type of cloud service.

- Availability: in cloud computing, the most important criteria for quality of service is the availability of service. Availability is the probability that the cloud infrastructure or service are up and running in the specific time of utilities of the service provided for in the SLA
- Scalability: cloud consumers pay for the service only as they use it. The cloud provider should facilitate the specific resources for ease of scaling up and down. With scalability, cloud consumers can maximize revenue and cloud providers are able to optimize resources effectively.
- A clear method for cost calculation: service consumers using cloud computing are willing to pay as they use, so an annual billing period or even monthly periods are not suitable for cloud computing. A cost calculation for resource reservation method is not a unique method for each type of cloud service. For example, the storage service can be billed based on the time and size of the user's data. On the other hand, cloud CRM may be billed based on the number of users.
- The configuration of service: in cloud computing, users deal with virtual machines and these VMs should be configured in a flexible manner to enable users to execute business processes with minimal need for managing the effort of the configuration.
- Security and privacy: the critical data of a business must be stored and transferred via secure channels. If security features are not guaranteed by cloud providers, business organizations may spend too much on operating their own data centres rather than switching to cloud providers.

III. SLA Frameworks

To the best of our knowledge, scientific research in the area of SLA and trust management does not investigate the new paradigm of outsourcing of services in a pay-as-you-use framework, which is called "Cloud Computing". The main specifications which are designed to describe the syntax of SLA are: 1) Web Service Agreement (WS-Agreement) [5]. 2) Web Service Level Agreement Language and framework (WSLA) [6]. A WS-Agreement is created by Open Grid Forum (OGF) in order to create an official contract between service consumers and service providers. This contract should specify the guarantees, the obligations and penalties in the case of violations. Also, the functional requirements and other specifications of services can be included in the SLA. There are three main sections for WS-Agreement: name, context, and terms. A unique ID and optional names of services are included in the name section. The information about service consumer and service provider, domain of service, and other

specifications of service is presented in the context section. Terms of services and guarantees are described with more details in the terms section. These types of online agreements were developed for use with general services. For cloud computing, service consumers lack more specific solutions for SLA to present the main parameters of the visualization environment; at the same time these solutions should be dynamically integrated with the business rules of cloud consumers. The other specification is WSLA, which was developed to describe services in three categories, which are: 1) Parties: in this section, information about service consumers, service providers, and agents are described. 2) SLA parameters: in this section the main parameters which are measurable parameters are presented in two types of metrics. The first is resource metrics, a type of metric used to describe service provider's resources as row information. The second one is composite metrics. This metrics is used to represent the calculation of the combination of information about a service provider's resources. The final section of the WSAL specification is Service Level Objective (SLO). This section is used to specify the obligations and all actions when service consumers or service providers do not comply with the guarantees of services.

The primary shortcoming of these approaches is that they do not provide dynamic negotiation, and various types of cloud consumers need a different structure of implementation of SLAs to integrate their own business rules with the guarantees that are presented in the targeted SLA. In this paper, we propose a basic architecture for developing the service level agreement contract between service consumers and other parties such as service providers and external agents. Two main categories of SLA metrics are presented. Performance metrics show the measurements of performance parameters in cloud computing data centres such as response time and CPU capacity. The other metrics is business metrics; the main measurements of business-related aspects presented by this type of metrics includes such things as service cost and billing methods.

IV. Conceptual SLA Framework for Cloud Computing

A. *SLA Metrics*

In our proposed framework, the SLA parameters are specified by metrics. These metrics define how cloud service parameters can be measured and specify values of measurable parameters. In the cloud computing architecture, there are four types of services which providers can present to consumers. These services are: infrastructure as a service (IasS), platform as a service, software as a service, and storage as a service. The proposed SLA metrics for cloud computing consider these four types of these services. For each part of the SLA we define the most important parameters that consumers can use to create a reliable model of negotiation with this service provider. We focus on the definition of these parameters, and in our future work, we will design and implement the proposed framework followed by simulation experiments in order to validate our framework.

SLA metrics for IasS:

Companies like amazon.com provide infrastructure as a service. Most of the consumers are confused as to which important parameter should be defined in the hardware part of the SLA. We list the most important parameters for consumers who are interested in using cloud as an infrastructure service.

Table 1. SLA metrics for IasS

Parameter	Description
CPU capacity	CPU speed for VM
Memory size	Cash memory size for VM
Boot time	Time for MV to be ready for use
Storage	Storage size of data for short or long term of contract
Scale up	Maximum of VMs for one user
Scale down	Minimum number of VMs for one user
Scale up time	Time to increase a specific number of VMs
Scale down time	Time to decrease a specific number of VMs
Auto scaling	Boolean value for auto scaling feature
Max number can be configured on physical server	Maximum number of VMs that can be run on individual server
Availability	Uptime of service in specific time
Response time	Time to complete and receive the process

SLA metrics for PasS:

Platform as a service is a type of cloud computing that provides all the requirements needed to support application developers in developing, evaluating, and delivering applications and software for end users [7]. So, in this case, developers using PasS do not need to download tools or configure hardware to complete the developing tasks. For SLA metrics related to PasS, we define the main parameters that can be used as basic criteria when developers want to negotiate with PasS providers.

Table 2. SLA metrics for PasS

Parameter	Description
Integration	Integration with e-services and other platforms
Scalability	Degree of use with large number of online users
Pay as you go billing	Charging based on resources or time of service
Environments of deployment	Supporting offline and cloud systems
Servers	
Browsers	Firefox, IExplorer,..
Number of developers	How many developers can access to the platform

SLA metrics for SasS:
Software as a service is a common example of cloud services [8] if an application is hosted on a cloud platform and infrastructure to provide built-in services for end users of cloud computing. Good examples of SasS are mail, calendar, and social web sites provided by Google, Yahoo, and Microsoft. We present the common metrics parameters for SasS as an example of metrics for this type of cloud service.

Table 3. SLA metrics for SasS

Parameter	Description
Reliability	Ability to keep operating in most cases
Usability	Easy built-in user interfaces
Scalability	Using with individual or large organisations
Availability	Uptime of software for users in specific time
Customizability	Flexible to use with different types of users

SLA metrics for Storage as a service:
Online users access their data from different geographical locations. In the past few years, online storage providers were unable to maintain large size of data because of the lack of huge space in storage disks, network performance, and data management systems. Now, data storage service providers such as S3 by amazon.com configure large numbers of storage hardware and they are able to manage and serve millions of users efficiently with their method of data transferral and ensuring these data are compatible with various types of applications. The parameters for data storage service metrics are basic requirements for negotiation with storage providers.

Table 4. SLA metrics for Storage as a service

Parameter	Description
Geographic location	Availability zones in which data are stored
Scalability	Ability to increase or decrease storage space
Storage space	Number of units of data storage
Storage billing	How the cost of storage is calculated
Security	Cryptography for storage and transferring of data, authentication, and authorization
Privacy	How the data will be stored and transferred
Backup	How and where images of data are stored
Recovery	Ability to recover data in disasters or failures
System throughput	Amount of data that can be retrieved from system in specific unit of time
Transferring bandwidth	The capacity of communication channels
Data life cycle management	Managing data in data centres, and use of network infrastructure

SLA general terms:
The above section presents the main parameters for metrics in four types of services. However, there are general metrics that can be defined for SLA with any or all types of cloud users. We present the most important parameters as an example when creating the basic SLA contract between cloud computing users and providers.

Table 5. SLA general terms

Term	Description
Monitoring	Who do the monitoring and what method of monitoring
Billing	Cost of service and how can be calculated
Security	Issues like cryptography, authentication, and authorization are main requirement for cloud users
Networking	The number of IPs, throughput, and load balancing
Privacy	How the data will be stored and transferred
Support service	Cloud providers should clearly define the methods of help and support
Local and international policies	The policy standards that providers follow

B. *Negotiation Strategies*

Negotiation is the method by which the service consumer and service provider present their terms and agree or disagree upon the results of this process to reach an agreement acceptable to both sides. There is more than one way of starting the negotiation process in an online environment [9, 10]. In this section, we discuss the possible negotiation scenarios relating to cloud computing. The first scenario involves direct negotiation between the cloud consumer and the cloud service provider. In this case, the service provider may create a unique template and define all SLA criteria such as period of contract, billing, and response time. When the SLA document is ready, cloud consumers can review the SLA terms and respond by signing the SLA, renegotiating or terminating the negotiation. Direct negotiation is a common method used by most of today's cloud providers. The second scenario is negotiation via a trusted agent, that is, an agent who has sound experience in selecting the cloud providers and defining the critical parameters for the SLA. This can be a key factor when a business wants to focus on the core business activities. A number of activities can be assigned to external agents who undertake the negotiation in flexible and reliable steps. They may start with the analysis of business processes and goals and complete the negotiation by monitoring all or some of the SLA parameters. Also, the trusted agent can use other agents to carry out some activities like service discovery and monitoring of performance. In the third scenario more than

one agent is used to carry out the one type of negotiation. As we mention above, there are four different types of cloud services: IasS, PasS, SasS, and storage as a service. A cloud consumer can sign a contract with four different agents (IasS agent, PasS agent, and SasS) which take the responsibility of defining SLA parameters and complete the negotiation process. This type of negotiation can be efficient if the cloud consumer requires more than one type of cloud service.

V. Conclusion and Future Work

The effective service level agreement is the key to ensure that a service provider delivers the agreed terms of services to the cloud consumer. In cloud computing, cloud consumers with clear definition of SLA parameters and flexible negotiation methods can increase the reliability and trust level of cloud provider-cloud consumer relationship. In this paper, the non-functional requirements of cloud consumers are presented and, based on these requirements, the most important criteria for the SLA are defined in order to help cloud users maintain a reliable protocol for negotiation with cloud service providers. The state-of-the-art SLA frameworks are discussed. Finally, we present three scenarios that can be applied to the cloud computing environment when consumers need to negotiate with cloud providers. As future work, we will design SLA metrics and implement a simulation process to test our framework in the cloud computing environment. The result of this work will be the basic tool to be used with trust management systems for cloud computing to help consumers select the most reliable service.

REFERENCES

[1] Armbrust, M., et al., Above the clouds: A Berkeley view of cloud computing. EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-28, 2009.

[2] Amazon. Amazon Elastic Compute Cloud (Amazon EC2). 2008 [cited; Available from: http://aws.amazon.com/ec2.

[3] Microsoft. Azure. [cited 2010 10 March]; available from: http://www.microsoft.com/windowsazure/.

[4] Google. Google Docs. [cited 2010 10 March]; available from: http://docs.google.com.

[5] Andrieux, A., et al. Web services agreement specification (WS-Agreement). 2004.

[6] Keller, A. and H. Ludwig, The WSLA framework: Specifying and monitoring service level agreements for web services. Journal of Network and Systems Management, 2003. 11(1): p. 57-81.

[7] Hilley, D., Cloud Computing: A Taxonomy of Platform and Infrastructure-level Offerings. 2009.

[8] Muller, J., et al. Customizing Enterprise Software as a Service Applications: Back-End Extension in a Multi-tenancy Environment. 2009: Springer.

[9] Pichot, A., et al. Dynamic SLA-negotiation based on WS-Agreement. 2008: Citeseer.

[10] Rubach, P. and M. Sobolewski, Dynamic SLA Negotiation in Autonomic Federated Environments. 2009.

A Trust-Evaluation Metric for Cloud applications

Mohammed Alhamad, Tharam Dillon, Elizabeth Chang

***Abstract*— Cloud services are becoming popular in terms of distributed technology because they allow cloud users to rent well-specified resources of computing, network, and storage infrastructure. Users pay for their use of services without needing to spend massive amounts for integration, maintenance, or management of the IT infrastructure. Before interaction occurs between cloud providers and users, trust in the cloud relationship is very important to minimize the security risk and malicious attacks. The notion of trust involves several dimensions. These dimensions include: the scalability, availability, security, and usability parameters of IaaS, PaaS, SaaS, and DaaS. Each of these dimensions is characterized by fuzzy aspects and linguistic terms. This paper develops a model for each of the dimensions for IaaS using fuzzy-set theory. It then uses the sugeno fuzzy-inference approach for developing an overall measure of trust value of the cloud providers. It is not easy to evaluate the cloud metrics for a general domain. So, in this paper, we will use an e-learning application as the main example when we collect the data and apply the fuzzy model to evaluate the trust for cloud computing. Test and results are presented to evaluate the effectiveness and robustness of the proposed model.**

***Index Terms*— Trust, Cloud computing, Fuzzy inference.**

I. Introduction

A. Cloud Computing

In the literature, various definitions of cloud computing have been proposed [1-3]. In this paper, we adopted and considered the definition provided by U.S. NIST (National Institute of Standards and Technology) that describes cloud computing as "... a model for enabling convenient, on demand network access to a share pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management afford or service provider interaction" [1]. In other words, cloud computing is a framework by means of which virtualized infrastructure resources are delivered as a service to customers by using a public network which is the Internet [4-6]. The cloud customers can range from big organizations, small business and developers to individual users. In this paper, we will refer to such customers as 'users'. One of the advantages of having such a framework is that users do not need to buy costly physical infrastructure or software, but they can obtain them over a virtual environment from other users at a much lower price, thereby reducing their operational and maintenance costs. For example, Salesforce.com developed a customer relationship management solution (CRM) and delivered this as a cloud service, not as a package of software. Salesforce.com customers can use this type of service using a basic machine with an Internet browser [7]. There are four main delivery models of cloud services with such a paradigm. They are:

1. Infrastructure as a service (IaaS): In such architectures, users can use the visualization resources as a fundamental infrastructure for their applications. These resources may be a CPU, network, or storage. Cloud users can manage the resources and assign rules for end users [8].
2. Database as a service (DaaS): Such architectures allow users to rent a specific size of storage for a specific period of time. Users are not required to manage the integration or the scaling of the infrastructure. Database providers take the responsibility for integration, privacy, and security of users' data [9].
3. Platform as a service (PaaS): In these architectures, users utilize all facilities on the cloud to develop and deliver their web application and services to the end users. PaaS services may include development, integration, testing or the storage resources to complete the life cycle of services [10].
4. Software as a service (SaaS): Here, users connect with the service providers to use the application, but they do not control the infrastructure, operating system or network infrastructure [10, 11].

Each of these delivery models is above the required hardware and virtualization model. No matter what type of delivery model is being used, there are five essential factors or characteristics that have to be satisfied in order to achieve smooth computing in a cloud computing environment. They are:

a) On-demand self-service: This refers to the availability of the required resources (such as CPU power, network etc) as and when the user needs it. Furthermore, this should be without any human intervention [12].

b) Broad network access: As the interacting medium between the different users is the Internet, there should be a broad network access available that allows for the seamless

Manuscript received July 10, 2011.

Mohammed Alhamad is with the Digital Ecosystems and Business Intelligence Institute (DEBII), Curtin University, (Mohammed.Alhamad@postgrad.curtin.edu.au).

Tharam Dillon is Distinguished Research Professor and Head of R&D of Digital Ecosystems and Business Intelligence (DEBII), Curtin University (Tharam.Dillon@cbs.curtin.edu.au).

Elizabeth Change is Director of the Research Institute for Digital Ecosystems and Business Intelligence (DEBII), Curtin University (Elizabeth.Chang@cbs.curtin.edu.au).

interaction of different applications across different heterogeneous platforms [13].

c) Resource Pooling: A cloud provider should support multi-tenancy of its resources for maximising the efficiency of its infrastructure. For example, it should be able to dynamically assign the required resources to the consumer according to its demand [14].

d) Rapid Elasticity: It should be flexible according to the computing resources required for the customers. For example, there is no up-front commitment and the customers should be able to relinquish the resources once their work is done [15].

e) Measure of Service: There should be a framework that measures the usage of each user according to the resources that are being used by it [16].

B. Trust and Fuzzy Inference System

Trust is a very important factor in the open distributed systems. It is the main concern of all interactions between service providers and consumers in such a changing environment. The trust evaluation process is not clear and easy for diverse users because it has vague and different subjective values. This leads to the need for a clear description methodology to present the values of trust in a clear way. Fuzzy logic is an effective technique for solving this problem. The fuzzy logic approach is a better method of describing human perception. Therefore, this paper will use this approach to evaluate the trustworthiness of service providers in cloud computing.

In this paper, we will develop an approach that characterizes the key aspects of the trust relationship between cloud providers and users. Moreover, each of the trust dimensions will be represented within a fuzzy framework, and measures along each dimension will be developed. In addition, an overall figure for trust value will be developed for the cloud providers.

C. E-learning systems

Electronic learning is an online service that includes a wide range of e-services and applications that provide different types of online-based media to deliver fast and effective training and education. The Internet is the main tool that e-learning providers use for the delivery of complicated functions for students and school staff. E-learning systems are widely used in various sectors of teaching which include universities, companies, medical organizations, and even in training schools at a lower level. The main stockholders involved in e-learning systems are the trainers and students. Students can derive major benefits from an e-learning environment in terms of efficient media delivery time, timely feedback, communication with teachers etc. Also, students can communicate interactively with tutors and students anywhere at any time. E-learning providers can obtain more benefits by using the cloud computing infrastructure. Most of the e-learning legacy systems can be adjusted and transferred to cloud data centres to provide more scalable and available applications for students who can be anywhere in the world. In this paper, we use the e-learning system as an example of cloud applications that can be assisted and evaluated using our proposed approach which is a fuzzy logic technique. We chose an e-learning application to save the data collection time; also, there are many experts available for consultation when we define the fuzzy inputs and outputs parameters.

II. RELATED WORK

Different models have been proposed with fuzzy logic to provide reliable and trusted solutions for online services. For instance, Falcon et al. [17] implemented a socio-cognitive model to evaluate the trustworthiness. A Fuzzy Cognitive Map is used with different components to evaluate the impact and can be changed to suit different situations. Another model proposed by Sabater and Sierra [18] uses a fuzzy logic-based approach to analyse the relationships of the service users in electronic marketplaces. Reputation mechanisms are used in e-market systems (e.g. Amazon, E-bay) to secure the transactions of all users in a centralized architecture. Novel models of reputation and trust have been developed in e-market places to provide reliable services of security since traditional solutions to security issues do not adequately protect providers and services consumers [19]. The most important aspect of these models is the information relating to past behaviours of users. This information is used to present the reputation of those users in terms of availability, reliability, and security. As centralized architectures of online reputation models, E-bay and Amazon exemplify this approach. Their systems are implemented based on a centralized rating model so that customers and sellers can rate each other using numerical ratings or feedback comments. Users can obtain a reputation profile for a given user in order to decide whether or not to proceed with a transaction with this user. For example, E-bay uses 1, 0, -1 scales which means positive, neutral, and negative respectively. Users use these scales to rate business partners based on past behaviours. The feedback from users is stored in a central system and the reputation score is computed regularly as cumulative results of user ratings [20]. The problem with this mechanism is that users with high scores for reputation can cheat other users in the course of a few transactions even though they receive negative feedback, because these users still receive positive ratings from other customers. Also, this model cannot guarantee the consistent performance of all services from the one user. This model employs a centralized architecture; therefore, all services and reputation information have a single point of failure. Unlike the centralized architecture of service discovery, the peer-to-peer model does not use a single point to manage and store descriptions of services and reputation information. Vu et al. [8] propose peer-to-peer web service discovery that uses QoS and users' feedback to rank and select services. QoS data about services and reputation rates from consumers are stored in multi-peers in peer-to-peer systems. Monitoring agents are used to prevent cheating by users and providers. Trusted agents monitor and provide reports of services to a UDDI peer and, based on this information, services are evaluated and ranked. However, the monitoring of reports differs from peer to peer, because each peer uses different criteria to provide feedback about services. Hence, it is evident that the proposed works in trust and reputation management systems are designed mainly to enhance the security of the traditional web services. In cloud computing, the execution of services has changed to be

completely independent of the consumer's infrastructure. The proposed methodology will present a novel metrics of trust for cloud computing providers. An e-learning application will be used to show the effectiveness of the proposed method when the performance evaluation is needed for cloud providers.

III. RESEARCH METHODOLOGY

The research approach includes problem definition, definition of domain variables, data collection, model design, and the implementation. Figure 1 describes the research approach. The following sections describe the research steps.

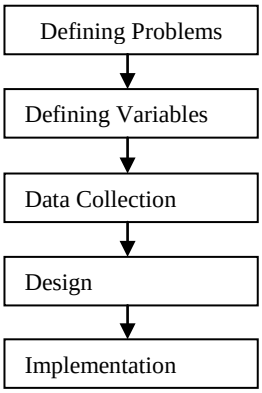


Figure. 1. The research approach.

A. Problem Definition

In the existing body of literature on cloud computing, there is no framework by which a cloud service consumer can make an intelligent trust-based decision regarding service selection from a service provider. Given the potential growth of cloud computing and the business implications, it is very important to have such architecture in place. The primary issues which are not investigated in the related literature are:

- the lack of a reliable model for trust and reputation specified for cloud architecture;
- the difficulties faced by cloud users when they want to sign online agreements with cloud providers; there is no clear and reliable method for selecting the most suitable parameters for the SLA contracts;
- the lack of a proposed model to calculate and estimate the cost for each level of the cloud architecture;
- although trust and reputation systems have been widely proposed and implemented for various types of online services, no such models have been proposed for cloud computing; cloud users also need such systems in order to select the most trustworthy of services that are already being offered by cloud providers.

In this paper, we will focus on how to evaluate the trusted cloud providers in such a way that users of cloud can easily understand and start to build a trusting relationship with service providers.

B. Defining Domain Variables

During the process of designing the SLA model for cloud computing in our previous work [21], various parameters were investigated for cloud services as shown in Figure 2.

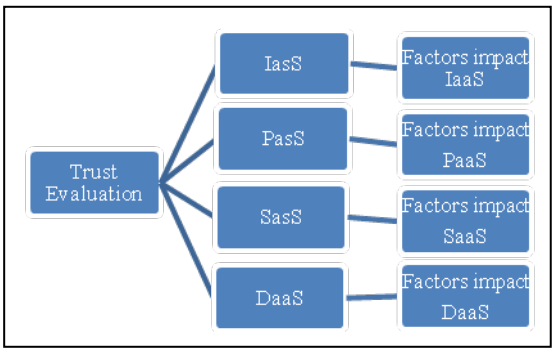


Figure. 2. Factors that impact on the different cloud services.

In this paper, we will use the main parameters which were designed for IaaS with additional parameters as the core factors that the cloud computing experts believe will impact on the trust evaluation of IaaS with an e-learning application. Table 1 lists the impact factors for trust evaluation of IaaS-based e-learning systems.

Table 1. Factors impact the trust value of IaaS-based e-learning systems

Final result of trust evaluation	**Factors impact trust**
Degree of trust (T)	Scalability (Sca)
	Availability (Avi)
	Security (Sec)
	Usability (Usa)

Trust evaluation factors include scalability, availability, security, and usability. After defining the problem, this section describes the inputs and output of the proposed model. This part of the research is usually conducted after investigating the problem domain and set more relationships with the cloud computing experts. We established good relationships with a number of cloud computing experts and end users and were able to define the most important variables for our model.

C. Data Collection

One of the most important steps in the development of fuzzy-based control systems is the data set preparation and collection. So, the model with a fuzzy inference approach must be trained with training data that represent the greatest possibilities of application [22]. In this study, we used the data which was collected from cloud computing experts and cloud users. An online-based survey was developed in order to collect more data sets from different locations. The survey with the designed research questions was conducted to collect values for the most important variables which had already been selected to present the trust value in a cloud-based e-learning application.

D. Design Fuzzy Model

Fuzzy logic theory is used so as to extend the mathematics ontology in a certain method with fuzziness in order to help make an intelligent decision. By basing our research on the application of fuzzy logic technique to the cloud services environment, we aim to solve the problem of uncertainty in the evaluation of trust for cloud providers. The proposed fuzzy logic method in this paper uses three fuzzy sets for the input factors and five fuzzy sets for the parameters of output. The three fuzzy sets which are low (L), medium (M) and high (H) are used to characterize the fuzzy value for each input which are scalability, availability, security and usability. The fuzzy sets that represent the output parameters are: very poor (VP), poor (P), good (G), very good (VG) and excellent (E).

The use of three fuzzy sets for four inputs as shown in Figure 3, will generate 241 fuzzy rules. This takes into consideration all possible combinations of inputs. This is a large number of rules and many of these rules are unnecessary when using the Sugeno fuzzy technique. Hence, the neural networks [23] will be used to reduce the

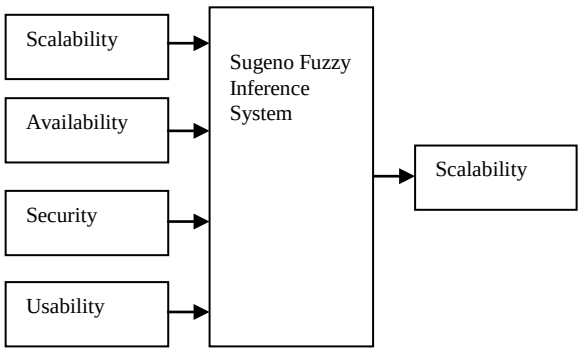


Figure. 3. Sugeno fuzzy inference model

number of fuzzy rules. So, the proposed model provides a more convenient method of evaluating the trust value of cloud providers. Table 2 shows the sample's fuzzy rules for input factors and the assigned values for output. The type of membership function for inputs and output that can be used depends on the nature of the system's attributes. In this paper, the gbell membership function is used because this is the simplest membership function that can present the input data and give a better view when we analyse the experiment results.

Table. 2. Samples of fuzzy rules for trust evaluation of IaaS

IF Sca	AND Avi	AND Sec	AND (Usa)	Then T
L	L	L	L	VP
M	L	L	M	P
M	M	L	M	G
L	M	M	L	P
M	L	M	M	G
H	L	L	H	P
M	M	M	M	G
H	H	L	H	G
L	H	H	L	G
H	L	H	H	VG
H	M	M	H	G
M	H	M	M	G
H	M	H	H	VG

The proposed system will help cloud users to make intelligent decision in simple method. Figure 4 explain the main process for trust decision making for cloud providers.

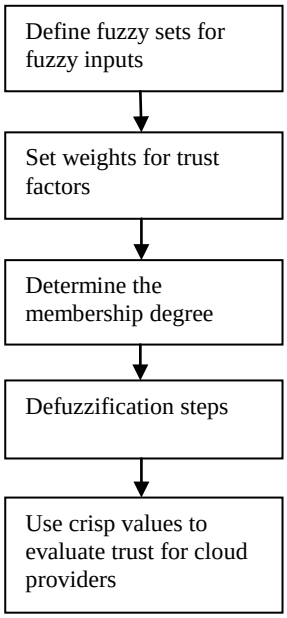
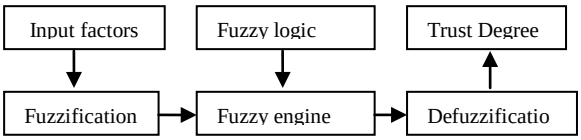


Figure. 4. Trust decision making process for cloud providers

The basic processes for the proposed model are explained as follows:

1) The first step of the proposed method is defining the fuzzy sets for all factors and fuzzy sets for the output which is the trust value for cloud providers. Then, requesters for trustworthiness about a service provider can set the weights for each factor; this step depends on the application of cloud services. If requesters do not like to provide these weights, the proposed fuzzy system will deal with all factors equally. Figure 5 describes the processes of the proposed system.



Fugure.5. Fuzzy model processes

2) The second step is the fuzzification. In this part of the process, all inputs are assigned to the appropriate degree of the fuzzy sets of input. This process uses a membership function to determine the degree of input to the fuzzy set.
3) Select the fuzzy membership function. Membership function is a function that determines how each of the values in the input range belong to the input space of the membership value. The membership range is between 0 and 1. Figure 6 presents one of the membership functions of the scalability factor.

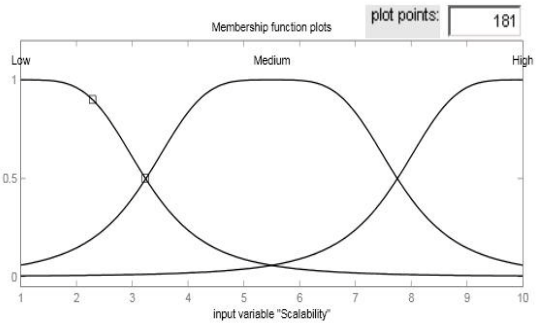


Figure. 6. Membership function of the scalability factor

4) Design fuzzy rules. In this paper, we select the most important rules of the inference system based on the neural networks [23]. Discussions and an online survey are used to determine the means of selecting fuzzy rules that give our system more accurate and better performance.
5) Sugeno fuzzy inference engine. The mean technique which is used in the proposed model is the Sugeno fuzzy inference method. Sugeno's method is one of the most popular control approaches which uses the fuzzy theory. The Sugeno fuzzy inference takes the fuzzy set of inputs to produce a final output value as a crisp value..
6) Defuzzification. This type of fuzzy method uses centroid calculation in the process of defuzzification to produce a single output value.

IV. EXPERIMENT

This section shows our methodology for verifying the proposed trust calculation model for cloud-based online services. In this section, the implementation of the proposed model is provided with the final results of the experiment. The fuzzy logic toolbox of Matlab is used to design and implement our model. This toolbox includes ready functions and calculation methods to implement more than one type of fuzzy inference systems such as the Mamdani and Sugeno inference system. In our model, we use the Sugeno fuzzy method with gbell membership function for inputs and output. Figure 7 presents the main model for the fuzzy logic system. We used FIS editor in Matlab to develop the model. The proposed model was implemented with four input factors: scalability, availability, security, and usability. These four inputs are directed as inputs to the fuzzy inference system implemented with the Sugeno method.

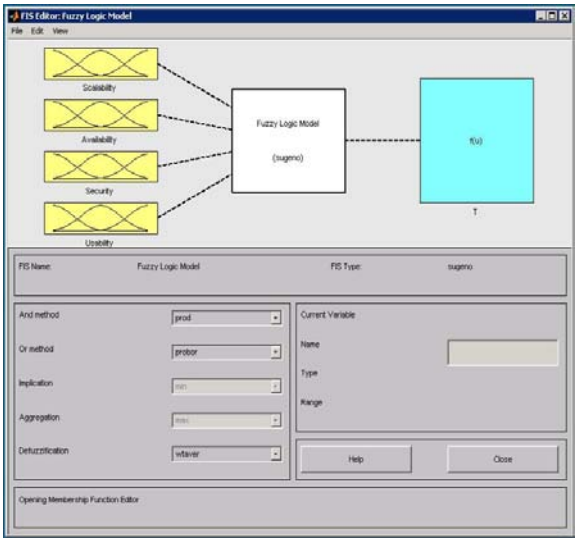


Figure.7. FIS editor interface.

Using the FIS editor, we have trained the system with 54 datasets among the 81 datasets which were collected for this experiment. Figure 8 shows the system after the training process.

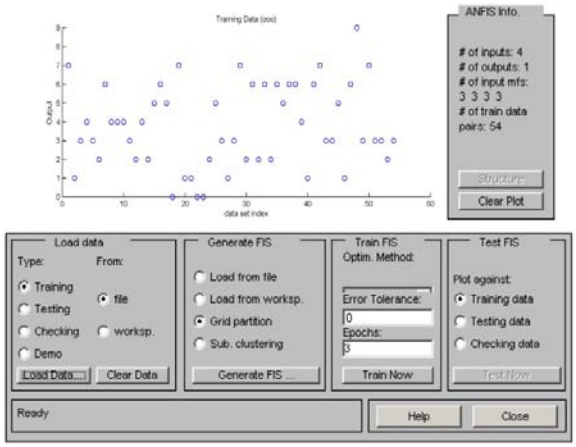


Figure. 9. Training data sets

In the training process, we undertake two main steps. The first step is to learn the model structure. IF-THEN rules and the knowledge from experts are used to learn the system in the first step. In the second step, we use membership functions and select the related rules to learn the parameters of inputs. Figure 10 shows the proposed system after the training process using two training steps.

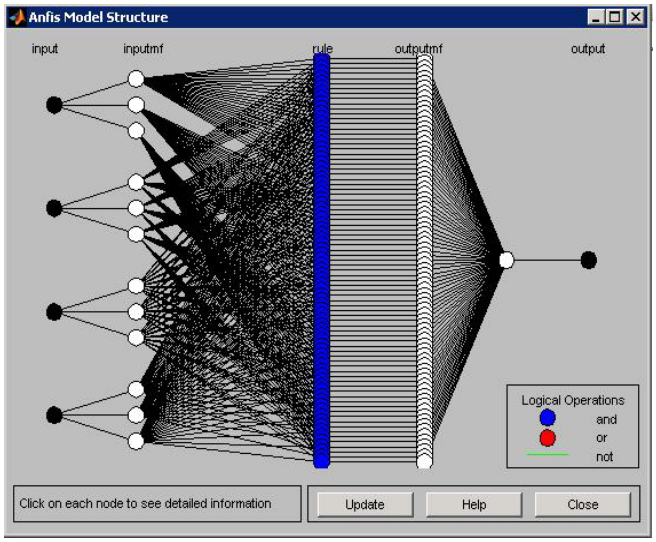


Figure. 10. Fuzzy inference system after training steps.

V. CONCLUSION

In this paper, a trust evaluation scheme based on fuzzy logic system is described. The proposed scheme enables cloud users to evaluate the trustworthiness of cloud services providers when creating or shifting their distributed systems to cloud data centres. Our evaluation method uses the cloud-based e-learning system with certain factors such as security and availability as an example of a cloud-based service. We believe the proposed model provides a valid method, since the obtained results of the experiments are close to the model output using the real data sets. This fuzzy-based model can be extended for use with additional input factors. Moreover, our model can also be extended to various web-based applications such as e-commerce applications etc.

REFERENCES

[1] P. Mell and T. Grance, "Draft nist working definition of cloud computing," Referenced on June. 3rd, 2009.

[2] J. Napper and P. Bientinesi, "Can cloud computing reach the top500?," 2009, pp. 17-20.

[3] Y. Chen, et al., "What's New About Cloud Computing Security?," 2010.

[4] R. Buyya, "Market-Oriented Cloud Computing: Vision, Hype, and Reality of Delivering Computing as the 5th Utility," 2009, p. 1.

[5] A. Marinos and G. Briscoe, "Community cloud computing," CoRR, abs/0907.2485, 2009.

[6] P. T. Jaeger, et al., "Cloud computing and information policy: Computing in a policy cloud?," Journal of Information Technology & Politics, vol. 5, pp. 269-283, 2008.

[7] M. Nelson, "Building an Open Cloud," Science, vol. 324, p. 1656, 2009.

[8] D. Hilley, "Cloud Computing: A Taxonomy of Platform and Infrastructure-level Offerings," 2009.

[9] H. Cai, et al., "Customer Centric Cloud Service Model and a Case Study on Commerce as a Service."

[10] D. Cerbelaud, et al., "Opening the clouds: qualitative overview of the state-of-the-art open source VM-based cloud management platforms," 2009, pp. 1-8.

[11] J. Muller, et al., "Customizing Enterprise Software as a Service Applications: Back-End Extension in a Multi-tenancy Environment," 2009, p. 66.

[12] B. Sotomayor, et al., "Virtual infrastructure management in private and hybrid clouds," IEEE Internet Computing, vol. 13, pp. 14-22, 2009.

[13] D. Nurmi, et al., "The eucalyptus open-source cloud-computing system," 2009, pp. 124-131.

[14] M. Zeller, et al., "Open standards and cloud computing: Kdd-2009 panel report," 2009, pp. 11-18.

[15] T. Dillon, et al., "Cloud Computing: Issues and Challenges," 2010, pp. 27-33.

[16] J. Nunamaker Jr, et al., "Systems development in information systems research," Journal of Management Information Systems, pp. 89-106, 1990.

[17] R. Falcone, et al., "A fuzzy approach to a belief-based trust computation," Trust, reputation, and security: theories and practice, pp. 55-60, 2003.

[18] J. Sabater and C. Sierra, "Reputation and social network analysis in multi-agent systems," 2002, pp. 475-482.

[19] T. Grandison and M. Sloman, "A survey of trust in internet applications," IEEE Communications Surveys and Tutorials, vol. 3, pp. 2-16, 2000.

[20] P. Resnick and R. Zeckhauser, "Trust among strangers in Internet transactions: Empirical analysis of eBay's reputation system," Advances in Applied Microeconomics: A Research Annual, vol. 11, pp. 127-157, 2002.

[21] A. Mohammed, et al., "Response Time for Cloud Computing Providers," presented at the IIWAS, Paris.

[22] C. Yin, et al., "Methods to improve prediction performance of ANN models," Simulation Modelling Practice and Theory, vol. 11, pp. 211-222, 2003.

[23] S. Sestito and T. Dillon, "Automated knowledge acquisition," 1994.

A Survey on SLA and Performance Measurement in Cloud Computing

Mohammed Alhamad, Tharam Dillon, Elizabeth Chang

Curtin University, Australia

Mohammed.Alhamad@postgrad.curtin.edu.au, Tharam.Dillon@cbs.curtin.edu.au,
Elizabeth.Chang@cbs.curtin.edu.au

Abstract. Cloud computing has changed the strategy used for providing distributed services to many business and government agents. Cloud computing delivers scalable and on-demand services to most users in different domains. However, this new technology has also created many challenges for service providers and customers, especially for those users who already own complicated legacy systems. This paper reviews the challenges related to the concepts of trust, SLA management, and cloud computing. We begin with a survey of cloud computing architecture. Then, we discuss existing frameworks of service level agreements in different domains such as web services and grid computing. In the last section, we discuss the advantages and limitations of current performance measurement models for SOA, distributed systems, grid computing, and cloud services. Finally, we summarize and conclude our work.

Keywords: SLA, Measurement, Cloud computing

1 Introduction

Cloud computing has been the focus of active and extensive research since late 2007. Before the term ‘cloud’ was coined, there was grid technology. Now, the hot topic of research is cloud and more proposed frameworks and models of various solutions for the new technology have started to be applied to the cloud architecture. In this section, we survey the literature in order to determine the most appropriate definition of “cloud computing”. Also, we review the different architectural frameworks and the common challenges that may present major problems for providers and customers who are interested in understadning this type of distributed computing.

The Google trends report shows that cloud computing had surpassed grid computing by late 2007.

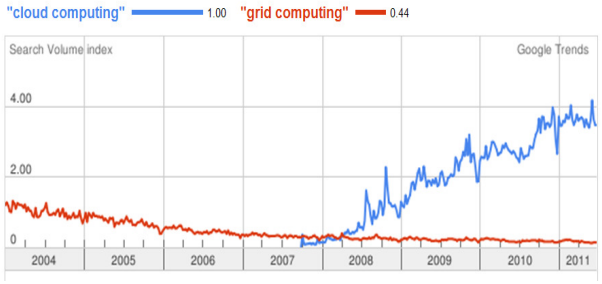


Fig. 1. Cloud computing trend, source, Google search engine

2 Definition

Experts and developers who investigate issues and standards related to cloud computing do not necessarily have the same technology background. In research projects, professionals from grid technology, SOA, business, and other domains of technology and management have proposed several concepts to define cloud computing. These definitions of cloud computing still need to be presented in a common standard to cover most technology and other aspects of cloud computing architecture.

In the context of networking and communication, the term “cloud” is a metaphor for the common internet concept [1]. The cloud symbol is also used to present the meaning of network connection and the way that the cloud technology is provided by internet infrastructure. “Computing” in the context of the cloud domain refers to the technology and applications that are implemented in the cloud data centers [2].

In [3], Vaquero et al. comment on the lack of a common definition of cloud computing. They state that developers and business decision makers confuse the understanding of the technology with the features of cloud data centers. So, large budgets may be allocated to implement private or even public cloud data centers. However, these data centers face several problems when users or public customers want to connect the interfaces of their legacy systems with the new technology of cloud architecture. Vaquero et al. link the challenge of maximizing the revenue of building cloud technology to professionals who are involved in distributed services. Because they come from a traditional computing domain, they are confused about the other concepts of distributed services such as grid and web services. The definition used in [3] is as follows:

"Clouds are a large pool of easily usable and accessible virtualized resources (such as hardware, development platforms and/or services). These resources can be dynamically re- configured to adjust to a variable load (scale), allowing also for an optimum resource utilization. This pool of resources is typically exploited by a pay-per-use model in which guarantees are offered by the Infrastructure Provider by means of customized SLAs".

Although, this definition presents the main features of cloud computing, it does not encompass other important components of cloud architecture which include the method of establishing and managing network, applications, and supporting services.

Wang [4] defines cloud computing as:

"*A computing Cloud is a set of network enabled services, providing scalable, QoS guaranteed, normally personalized, inexpensive computing infrastructures on demand, which could be accessed in a simple and pervasive way".*

Wang's definition of cloud focuses on the technical aspects of services. Business and functional characteristics are absent from the proposed definition. On other hand, Gruman and Knorr [5] explain the main technical concepts of a cloud services model and define cloud computing from the developers' perspective. The authors show how the cloud computing architecture takes advantage of the way that different distributed services (mainly web services and SOA) are implemented. Two types of cloud services are presented along with this definition; they define SaaS and PaaS. Despite the importance of IaaS as a main component of cloud architecture, they do not adequately discuss this type of cloud delivery model.

In this paper, we adopted and considered the definition provided by U.S. NIST (National Institute of Standards and Technology) [6], according to which "Cloud computing is a model for enabling convenient, on demand network access to a share pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management afford or service provider interaction" [6].

The shortcomings of the proposed definitions of cloud computing given above are as follows:

1. None of the definitions consider cloud computing from the technical and business perspectives. This would cause confusion to decision makers in large organizations, especially when they want to define the parameters of a costing model of cloud services.

2. Existing cloud definitions do not specify the onus of responsibility in cases of poor QoS delivery.

3. Most of the proposed definitions consider specific types of cloud services, whereas a comprehensive definition of cloud should clearly define all classes of cloud services.

4. The proposed definitions do not consider a definition of cloud users.

The following tables include the scope of definitions and list the main shortcomings of the definitions discussed above:

Table 1. Conclusion of cloud definitions

Reference	**The scope of definition**	**Missing**
Vaquero [3]	Define architecture and service model	Management, supporting, and trust concepts
Wang [4]	Technical concepts	Business and functional characteristics
Gruman [5]	Comparison between cloud computing and web services, and SOA	Definition of IaaS, and DaaS
Mell [6]	Technical features, management, and security concepts	Costing and billing model

3 Taxonomy of Cloud Computing

Buyya [7] presents more than fifteen characteristics to distinguish cloud computing from other distributed systems. Buyya uses scalability, automatic adaption, virtualization, and dynamic model of billing as main concepts to construct the architecture of cloud computing. Moreover, he explains how cloud services can be delivered to different types of users. For instance, users who want to develop small size applications may connect to one of the PaaS such as Microsoft Azure [8] without the need to install any of the development tools. Hoefer [9] identifies a clear taxonomy framework for existing categories of cloud services. The class of cloud services is described in a tree-structured taxonomy, and the unique characteristics of each model of service are used to identify each node of the proposed tree structure. Hoefer's classification provides a clear comparison of cloud services at high level on the tree structure. However, at the base of the structure, the taxonomy of cloud services is not enough to distinguish services in more detail. The taxonomy presented by Laird in [10] defines cloud technology from the perspective of service providers. The proposed taxonomy presents the common vendors of cloud services. Laird presents two classifications of services. The first class defines the infrastructure of cloud services, and the second one defines the services based on cloud features such as security, billing, and applications which are built into the system. Rimal [11] presents a comprehensive framework for the architecture of cloud computing. He describes the taxonomy of cloud services with more focus on the management domain of cloud contents. The concepts of management, business, billing, and support of cloud services are defined in great depth in order to present the cloud architecture as a new business model. The main advantage of the proposed work by Rimal is that relationships between security features and cloud components are provided as a part of the comparison of service models in cloud computing. The taxonomy proposed by Oliveira [12] classifies the concepts of cloud computing according to dimensions of cloud architecture, business model, technology infrastructure, pricing, privacy, and standards. The proposed taxonomy is provided in a hierarchical tree with parent and child relationships. Oliveira uses SaaS, PaaS, IaaS, and DaaS as sub-taxonomy for the business model. This classification is used in the literature of cloud computing to distinguish the service delivery for end users of cloud services. These sub-taxonomy terms may cause confusion in understanding the way that various business models are constructed for cloud services. The taxonomy proposed by Oliveira describes the concepts of cloud architecture from the perspective of e-science. Therefore, many of the technical aspects of cloud computing are missing from the proposed taxonomy.

2 Service Level Agreements

A service level agreement is a document that includes a description of the agreed service, service level parameters, guarantees, and actions for all cases of violation. The SLA is very important as a contract between consumer and provider. The main idea of SLAs is to give a clear definition of the formal agreements about service terms like performance, availability and billing. It is important that the SLA include the obligations and the actions that will be taken in the event of any violation, with clearly expressed and shared semantics between each party involved in the online contract.

This section discusses works related to SLAs in three domains of distributed services. Firstly, we discuss the proposed SLAs structure for web services. Secondly, the frameworks of SLAs designed to grid computing are reviewed; thirdly, we discuss the main works that specifically focus on cloud computing. Finally, we include in this section the main shortcomings of these SLA frameworks.

A) SLAs for Web Services

Several specifications for defining SLAs have been proposed for web services. WSLA language [13] introduces a mechanism to help users of web services to configure and control their resources in order to meet the service level. Also, the service users can monitor SLA parameters at run time and report any violation of the service. WSLA was developed to describe services under three categories: 1) Parties: in this section, information about service consumers, service providers, and agents are described. 2) SLA parameters: in this section the main parameters which are measurable parameters are presented in two types of metrics. The first is resource metrics, a type of metrics used to describe a service provider's resources as row information. The second one is composite metrics. This metrics is used to calculate the combination of information about a service provider's resources. The final section of the WSAL specification is Service Level Objective (SLO). This section is used to specify the obligations and all actions when service consumers or service providers do not comply with the guarantees of services. The WSLA provides an adequate level of online monitoring and contracting, but does not clearly specify when and how a level of service can be considered a violation. WSOL [14] is a service level specification designed mainly to specify different objectives of web services. Defining concepts of service management, cost and other objectives of services can be presented in WSOL. However, WSOL cannot adequately meet the objectives of the new paradigm of cloud computing.

WS-Agreement [15] is created by an Open Grid Forum (OGF) in order to create an official contract between service consumers and service providers. This contract

should specify the guarantees, the obligations and penalties in the case of violations. Also, the functional requirements and other specifications of services can be included in the SLA. The WS-Agreement has three main sections: name, context, and terms. A unique ID and optional names of services are included in the name section. The information about service consumer and service provider, domain of service, and other specifications of service are presented in the context section. Terms of services and guarantees are described in greater detail in the terms section. These types of online agreements were developed for use with general services. For cloud computing, service consumers need more specific solutions for SLAs in order to reflect the main parameters of the visualization environment; at the same time, these SLA solutions should be dynamically integrated with the business rules of cloud consumers.

The primary shortcomings of these approaches is that they do not provide for dynamic negotiation, and various types of cloud consumers need a different structure for the implementation of SLAs to integrate their own business rules with the guarantees that are presented in the targeted SLA.

B) SLAs for Grid Computing

In the context of grid computing, there are a number of proposed specifications which have been developed especially to improve security and trust for grid services. In [16], an SLA-based knowledge domain has been proposed by Sahai to represent the measurable metrics for business relationships between all parties involved in the transaction of grid services. Also, the author proposed a framework to evaluate the management proprieties of grid services in the lifecycle. In this work, business metrics and a management evaluation framework are combined to produce an estimated cost model for grid services. In our research, we extend this approach in order to build a general costing model based on the technical and business metrics of the cloud domain. The framework proposed in this work lacks a dynamic monitoring technique to help service customers know who takes responsibility when a service level is not provided as specified in SLA documents. Leff [17] conducted a study of the main requirements to define and implement SLAs for the grid community. The author provides an ontology and a detailed definition of grid computing. Then, a scientific discussion is presented about the requirements that can help developers and decision makers to deploy trusted SLAs in a grid community. A basic prototype was implemented in order to validate the use of SLAs as a reliable technique when the grid service provider and customer need to build a trusting relationship. The implementation of the framework in this study does not consider important aspects of security and trust management in grid computing. Keung [18] proposed an SLA-based performance prediction tool to analyse the performance of grid services. Keung uses two sources of information as the main inputs for the proposed model. The

source code information and hardware modelling are used to predict the value of performance metrics for grid services. The model proposed by Keung can be used in other types of distributed computing. But in the cloud environment, this model cannot be integrated with a dynamic price model of cloud services. It needs to be improved by using different metrics for cost parameters to reflect the actual price of cloud services. The system proposed by Padget in [19] considers the response time of applications in the grid systems. The main advantage of the proposed system is that it can predict the CPU time for any node in the grid network before conducting the execution. When Padget tested the adaptation SLA model using a real experiment on the grid, the prediction system produced values for response time close to the values obtained when users executed the same application on the grid. Noticing the delay recorded for the large size of executed files, the author claims that the reason for this delay is the external infrastructure such as internet connections. The author also discusses the impact of the time delay caused by external parties to the reputation of service providers when using SLA management systems. Although the author provides a good method for calculating the response time for grid resources, other metrics such as security and management metrics, are absent in this work.

C) SLAs for Cloud Computing

The context of this research is the management of service level agreements in cloud communities. In the sections above, we presented the frameworks and models in the current literature that are designed mainly for managing SLAs in traditional distributed systems. In this section, SLAs and approaches to agreement negotiations in the cloud community are presented.

Valdimir [20] describes the quality of services related to cloud services and different approaches applied to map SLA to the QoS. Services ontology for cloud computing is presented in order to define service capabilities and the cost of service for building a general SLAs framework. The proposed framework does not consider all types of cloud services; it is general and was tested on the Amazone EC2 only. It also needs to consider other types of cloud providers such as PaaS, DaaS, and SaaS. Our framework in this research considers this issue in the validation phase of the research. The framework developed by Hsien [21] focuses on software as a service model of delivery in cloud computing. More details are provided on how the services can be integrated to support the concept of stability of cloud community especially for SaaS.

The Shortcomings of the Proposed Works for SLAs in the Context of Distributed Services

The frameworks and structures that were discussed in previous sections have the following problems:

1. The existing frameworks focus more on the technical attributes than on the security and management aspects of services.

2. The proposed structures of SLAs in the above domains do not include a clear definition of the relationship between levels of violation and the cost of services.

3. Most of the above studies do not integrate a framework of trust management of the service provider with the collected data from monitoring systems of SLAs.

4. The concepts and definitions of service objectives and service descriptions included in SLAs are not easy to understand, especially for business decision makers.

5. The proposed works for cloud environments focus more on the evaluation of virtualization machines on local servers than on existing cloud service providers.

6. Most of the proposed structures of SLAs are defined by technical experts.

4 Performance Measurements Models

Cloud providers have been increased to deliver different models of services. These services are provided at different levels of quality of services. Cloud customers need to have a reliable mechanism to measure the trust level of a given service provider. Trust models can be implemented with various measurement models of services. As a part of this research, we investigate the use of a measurement approach in order to develop a general trust model for cloud community. In this section, the measurement model of SOA, distributed, and grid services will be reviewed.

A) SOA Performance Models

Kounev et al. in [22] propose an analytical approach to modelling performance problems in SOA-based applications. The authors discuss the different realistic J2EE applications for large systems of SOA architecture. A validated approach has been

tested for capacity planning of the organizations that use distributed services as an outsourcing infrastructure. The advantage of the proposed method is its ability to predict the number of application servers based on the collected information of SLA metrics. Walter et al. [23] implemented a simulation tool to analyse the performance of composite services. Authors used an online book store as a case study to simulate experiment scenarios. They focus on measuring communication latency and transaction completion time. Real data sets were compared with the simulation results. The authors state that the simulation tool presents results that approximate those of the real data. This type of simulation can be extended and applied to other distributed services. For cloud computing, more efforts is required to make this technique compatible with existing interfaces of cloud providers. Rud et al. in [24] use the WS-BPEL composition approach to evaluate the performance of utilization and throughput of SOA-based systems in large organizations. They developed the proposed methodology using a mathematical model in order to improve the processes of service level agreements in the SOA environment. The main focus of Rud's method is on the management aspects of services. However, this approach does not consider performance issues of response time, data storage, and other metrics of technical infrastructure. For the optimization of total execution time and minimization of business processes cost, Menasce in [25] provides an optimized methodology based on the comparison of performance metrics of SOA-based services. In this study, Menasce developed the proposed method to estimate the cost level of all services which are registered in the SOA directory under medium sized organizations. Then, the cost metric is compared to the real performance of services. The parameters of the performance metrics can be selected by service customers. So, the proposed model can be used for different types of services. Although, the proposed method produces a high level of reliability and usability, issues such as risk management, and trust mechanisms of the relationship between service providers and service customers are not discussed in more details.

B) Distributed Systems Performance Models

Kalepu et al. [26] propose a QoS-based attribute model to define the non-functional metrics of distributed services. Availability, reliability, throughput, and cost attributes are used in their work to define performance of resources of a given service provider. Two approaches of resources are used to calculate the final value of reputation. The first resource is the local rating record. Ratings of services which are invoked by local customers are stored in this record. In the second resource, global ratings of all services that are executed on resources of a given service provider are stored. Although, Kalepu et al. discuss the need to use SLA parameters to calculate the value of performance metrics, they do not explain how these parameters can be linked to the

local global resources of a rating system. In [27], Yeom et al. provide a monitoring methodology of the performance parameters of service. The proposed methodology uses the broker monitoring systems to evaluate the performance of resources of a service provider. Collected data of performance metrics are not maintained on the service consumer database. This method incurs low cost in terms of implementing measurement architecture but more risk in terms of privacy, availability of data, and security. Such risks are not easy to control, especially in the case of multi tenant distributed systems. Kim et al. in [28] analyse the quality factors of performance level of services and propose a methodology to assign priorities message processing of distributed web services based on the quality factors of services. This assigning aspect of their framework is a dynamic process in different service domains. They claim that their framework satisfies the agreement regarding service level in web services. The validation methodology of the proposed work lacks a clear definition of the evaluation criteria and a description of the way in which the experiment was conducted to produce the claimed results. The work proposed by Guster et al. in [29] provides an evaluation methodology for distributed parallel processing. In the proposed method, authors use a parallel virtual machine (PVM) and real hosting servers to compare the results of their experiments. The efficiency of the evaluation method performed better in PVM for the processing time. In the real server environment, the experiments presented better performance in terms of communication time. The evaluation of this work does not include the implementation processes and the experiment results are not clearly explained.

C) Cloud Computing Performance Models

Several studies already exist on the scalability of virtual machines. Most of these studies considered the measurement of performance metrics on the local machines. The background loads of tested machines are controlled to compare the results of performance with a different scale of loads. Evangelinos and Hill [30] evaluated the performance of Amazon EC2 to host High Performance Computing (HPC). They use a 32-bit architecture for only two types of Amazon instances. In our study, we run various experiments on most types of Amazon EC2 instances. These instances are: small, large, extra large, high CPU, medium, and high CPU extra large instance. Jureta, and Herssens [31] propose a model called QVDP which has three functions: specifying the quality level, determining the dependency value, and ranking the quality priority. These functions consider the quality of services from the customers' perspective. However, the performance issues related to cloud resources are not discussed and details are missing regarding the correlation of the quality model with the costing model of services. Cherkasova and Gardner [32] use a performance benchmark to analyse the scalability of disk storage and CPU capacity with Xen Virtual Machine Monitors. They measure the performance parameters of visualization

infrastructure that are already deployed in most data centres. But they do not measure the scalability of cloud providers using the visualization resources. However, our proposed work profiles the performance of virtualization resources that are already running on the infrastructure of existing cloud providers.

The Shortcomings of the Proposed Works for Above Performance models

1. The above proposed models for evaluating the virtualization services focus on how to measure the performance of virtual machines using local experiments. However, the techniques used for measuring the actual resources of cloud providers need further refinement in order to ensure some level of trust between service providers and the customers.
2. Most of the proposed works on performance evaluation do not allow service customers to specify the parameters of performance metrics. In cloud computing, service customers need a more flexible and dynamic approach to modify the parameters of performance metrics in order to solve the problem of dynamic changes of service requirements and business models of customers.
3. The experiments using the above proposed models do not specify the benchmarks for the performance evaluation.
4. In cloud computing architecture, the relationship between performance monitoring and costing metric is very important. The proposed models do not link the results of performance monitoring with the actual cost metric of services. So, service customers are not able to build a trust relationship with service providers without having a real cost model of services

References

[1] H. Katzan Jr, "On An Ontological View Of Cloud Computing," *Journal of Service Science (JSS),* vol. 3, 2011.

[2] D. C. Wyld, *Moving to the cloud: An introduction to cloud computing in government*: IBM Center for the Business of Government, 2009.

[3] L. M. Vaquero*, et al.*, "A break in the clouds: towards a cloud definition," *ACM SIGCOMM Computer Communication Review,* vol. 39, pp. 50-55, 2008.

[4] L. Wang, *et al.*, "Cloud computing: a perspective study," *New Generation Computing,* vol. 28, pp. 137-146, 2010.
[5] E. Knorr and G. Gruman, "What cloud computing really means," *InfoWorld,* vol. 7, 2008.
[6] P. Mell and T. Grance, "Draft nist working definition of cloud computing," *Referenced on June. 3rd,* 2009.
[7] R. Buyya, *et al.*, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems,* vol. 25, pp. 599-616, 2009.
[8] R. Jennings, *Cloud Computing with the Windows Azure Platform*: Wrox, 2010.
[9] C. Hoefer and G. Karagiannis, "Taxonomy of cloud computing services," 2010.
[10] P. Laird, "Different strokes for different folks: a taxonomy of cloud offerings," *Enterprise cloud submit, INTEROP,* 2009.
[11] B. P. Rimal, *et al.*, "A Taxonomy, Survey, and Issues of Cloud Computing Ecosystems," *Cloud Computing,* pp. 21-46, 2010.
[12] D. Oliveira, *et al.*, "Towards a Taxonomy for Cloud Computing from an e-Science Perspective," *Cloud Computing,* pp. 47-62, 2010.
[13] H. Ludwig, *et al.*, "Web service level agreement (WSLA) language specification," *IBM Corporation,* 2003.
[14] V. Tosic, *WSOL Version 1.2*: Carleton University, Dept. of Systems and Computer Engineering, 2004.
[15] A. Andrieux, *et al.*, "Web services agreement specification (WS-Agreement)," 2004.
[16] A. Sahai, *et al.*, "Specifying and monitoring guarantees in commercial grids through SLA," 2003.
[17] A. Leff, *et al.*, "Service-level agreements and commercial grids," *IEEE Internet Computing,* vol. 7, pp. 44-50, 2003.
[18] H. N. L. C. Keung, *et al.*, "Self-adaptive and self-optimising resource monitoring for dynamic grid environments," 2004, pp. 689-693.
[19] J. Padgett, *et al.*, "Predictive adaptation for service level agreements on the grid," *International Journal of Simulation: Systems, Science and Technology,* vol. 7, pp. 29–42, 2006.

[20] V. Stantchev and C. Schröpfer, "Negotiating and enforcing qos and slas in grid and cloud computing," *Advances in Grid and Pervasive Computing,* pp. 25-35, 2009.
[21] C. H. Wen*, et al.*, "A SLA-based dynamically integrating services Saas framework," pp. 306-311.
[22] S. Kounev and A. Buchmann, "Performance modeling and evaluation of large-scale J2EE applications," 2003, pp. 273-284.
[23] A. Walter and D. Potter, "COMPOSITION, PERFORMANCE ANALYSIS AND SIMULATION OF WEB SERVICES," 2007.
[24] D. Rud*, et al.*, "Performance modeling of ws-bpel-based web service compositions," 2006, pp. 140-147.
[25] D. A. Menascé*, et al.*, "A heuristic approach to optimal service selection in service oriented architectures," 2008, pp. 13-24.
[26] S. Kalepu*, et al.*, "Verity: a QoS metric for selecting Web services and providers," 2003, pp. 131-139.
[27] G. Yeom and D. Min, "Design and implementation of web services qos broker," 2005.
[28] D. Kim*, et al.*, "Improving Web services performance using priority allocation method," 2005.
[29] D. Guster*, et al.*, "Computing and network Performance of a distributed parallel processing environment using MPI and PVM communication methods," *Journal of Computing Sciences in Colleges,* vol. 18, pp. 246-253, 2003.
[30] C. Evangelinos and C. Hill, "Cloud Computing for parallel Scientific HPC Applications: Feasibility of running Coupled Atmosphere-Ocean Climate Models on Amazon's EC2," *ratio,* vol. 2, p. 2.34, 2008.
[31] I. Jureta*, et al.*, "A comprehensive quality model for service-oriented systems," *Software Quality Journal,* vol. 17, pp. 65-98, 2009.
[32] L. Cherkasova and R. Gardner, "Measuring CPU overhead for I/O processing in the Xen virtual machine monitor," 2005, p. 24.