A Survey on SLA and Performance Measurement in Cloud Computing

Mohammed Alhamad, Tharam Dillon, and Elizabeth Chang

Curtin University, Australia
Mohammed.Alhamad@postgrad.curtin.edu.au,
{Tharam.Dillon,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 understanding this type of distributed computing.

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

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011, Part II, LNCS 7045, pp. 469–477, 2011.

common standard to cover most technology and aspects of cloud computing. 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.

In this paper, we adopted and considered the definition provided by U.S. NIST (National Institute of Standards and Technology) [4], 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” [4].

Shortcomings of the proposed definitions of cloud computing 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.

3 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 [5] 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 [6] 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 [7] 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 [8], 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 [9] 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 [10] 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 [11] 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 [12] 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 [13] 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.

Shortcomings of the Proposals 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 [14] 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. [15] 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 [16] 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 [17] 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. [18] 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 [19], 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 [20] 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 [21] 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 [22] 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 [23] 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 [24] 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

5 Conclusions

The above discussions have highlighted many issues both in the development of SLAs and Performance Models for Cloud Computing which constitute rich fields for future research.

References

[1] Katzan Jr., H.: On An Ontological View of Cloud Computing. Journal of Service Science (JSS) 3 (2011)

[2] Wyld, D.C.: Moving to the cloud: An intro. to cloud computing in government. IBM Center for the Bus. of Government (2009)

[3] Vaquero, L.M., et al.: A break in the clouds: towards a cloud defin. ACM SIGCOMM Comp. Comm. Rev. 39, 50–55 (2008)

[4] Mell, P., Grance, T.: Draft nist working definition of cloud computing (referenced on June 3, 2009)

[5] Ludwig, H., et al.: Web service level agreement (WSLA) language specification. IBM Corporation (2003)

[6] Tosic, V.: WSOL Version 1.2: Carleton Univ., Dept. of Systems and Comp. Eng. (2004)

[7] Andrieux, A., et al.: Web services agree. spec. (WS-Agreement) (2004)

[8] Sahai, A., et al.: Specifying and monitoring guarantees in commercial grids through SLA (2003)

[9] Leff, A., et al.: Service-level agreements and commercial grids. IEEE Internet Computing 7, 44–50 (2003)

[10] Keung, H.N.L.C., et al.: Self-adaptive and self-optimising resource monitoring for dynamic grid environ., pp. 689–693 (2004)

[11] Padgett, J., et al.: Predictive adaptation for service level agreements on the grid. International Journal of Simulation: Systems, Science and Technology 7, 29–42 (2006)

[12] Stantchev, V., et al.: Neg. and enforcing qos and slas in grid and cloud comp. Adv. in Grid and Perv. Comp., 25–35 (2009)

[13] Wen, C.H., et al.: A SLA-based dynamically integrating services Saas framework, pp. 306–311

[14] Kounev, S., Buchmann, A.: Performance modeling and evaluation of large-scale J2EE applications, pp. 273–284 (2003)

[15] Walter, A., Potter, D.: Compos., Performance Analy. and Simulation of Web Services (2007)

[16] Rud, D., et al.: Performance modeling of ws-bpel-based web service compositions, pp. 140–147 (2006)

[17] Menascé, D.A., et al.: A heuristic approach to optimal service selection in service oriented architectures, pp. 13–24 (2008)

[18] Kalepu, S., et al.: Verity: a QoS metric for selecting Web services and providers, pp. 131–139 (2003)

[19] Yeom, G., Min, D.: Design and implementation of web services qos broker (2005)

[20] Kim, D., et al.: Improving Web services performance using priority allocation method (2005)

[21] Guster, D., et al.: Computing and netw. Perform. of a distr. parallel processing environ. using MPI and PVM commun. methods. J. Computing Sci. in Colleges 18, 246–253 (2003)

[22] Evangelinos, C., et al.: Cloud Comput. for paral. Sci. HPC Applic. Feasib. of run. Coup. Atmos. Ocean Climate Models on Amazon's EC2. Ratio 2, 2.34 (2008)

[23] Jureta, I., et al.: A comprehensive quality model for service-oriented systems. Software Qual. Journal 17, 65–98 (2009)

[24] Cherkasova, L., Gardner, R.: Measur. CPU overhead for I/O processing in the Xen virtual machine monitor, p. 24 (2005)