

The SLA Evaluation Model for Cloud Computing

Wu Chenkang¹, Zhu Yonghua², Pan Shunhong²

¹School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China

²Computing Center, Shanghai University, Shanghai 200444, China

raywck@shu.edu.cn, zyh@shu.edu.cn, pansh@shu.edu.cn

Abstract - Cloud Computing is an emerging technology to deliver the infrastructure, network, software, and even the development environment as a service in the philosophy of pay-as-you-use. To guarantee all participants' rights in the cloud, SLA is suggested to be a necessary contract which can protect users and service providers from violation. However, SLA is an early-proposed method that not suitable for new characteristics of cloud computing. This paper present an optimized SLA model for cloud computing and a corresponding evaluation method, which can determine a more fair and reliable evaluation result of cloud service. With this model, users can simplify their selection process, while providers can use the evaluation result for further resource allocation strategy.

Index Terms – SLA, Cloud computing, Evaluation model.

1. Introduction

Since Google first proposed the concept of “Cloud Computing”, this technology has gained great attention for its conveniences, flexibility and scalability. This online services integrate all kinds of resources which cloud providers have, and it has reduces the cost of purchasing, managing and maintaining the infrastructure which should have done by user himself. However, the development of cloud computing is restricted by how to constrain the responsibility and obligation of each participant in cloud circumstance.

It is important that cloud users should enjoy the resources and service level which guaranteed by cloud services providers, while the latter need to describe its services more precise to optimize the infrastructure of their own. One solution that is generally adopted is using Service Level Agreement (SLA), which is a contract signed between cloud providers and cloud users in order to provide various of details relating to service level. Long before cloud computing, SLA and its evaluation methods had already appeared, when most researches concerned about business or Service-oriented Architecture (SOA), such as E-business and Web Service. But these models and evaluation methods can no longer adapt to the new delivery model of cloud computing. So we proposed a novel SLA evaluation model for cloud computing to solve the problem.

The main contributions of this paper are summarized as follows:

1) Improve the classification method of the SLA model for cloud computing.

2) Proposed a unified dynamic evaluation method for different types of cloud services, which is a basis for both users selecting cloud service and providers scheduling resources.

The rest of the paper is organized as follows: Section 2 describes the definition of the SLA and its development. An improved SLA evaluation model for cloud computing is proposed in Section 3. In Section 4, the interests of both participants in cloud computing are discussed and a novel SLA evaluation method is developed as result. Section 5 concludes the paper.

2. Related Work

A Service Level Agreement is a contract signed between cloud service providers and users through negotiations, which represents the consensus of both sides in service, billing and remedies for violations [1]. With SLA, the quality of services is ensured. Also, the SLA helps to protect interest all participants.

The SLA has been used since late 1980s by telecom operators as part of the contract with their customers [2]. Then it becomes a standard protocol of business applications and Web Services. In current researches, there are two main specifications designed to describe the syntax of SLA:

1) *WSLA*: A language proposed by IBM, which describes the service respecting to parties involved in the negotiation, service level specification and service features [3]. The service level specification is presented by SLA parameters. The service features use Service Level Objective to indicate the responsibility when providers or users violate the agreed terms.

2) *WS-Agreement*: It is an alternative to WSLA created by Open Grid Forum to establish formal contracts between users and service providers in 2005 [4]. The SLA described by WS-Agreement is more capable of extending and expression, while improving the standardization, for it covers almost all characteristics of WSLA and using an XML-based negotiation document.

Web applications based on SOA architecture requires users selecting service only once. While in cloud computing, users need to submit their own resource/service request dynamically. As a result, it usually takes several rounds of negotiation to determine an SLA agreement. Neither in the above two specifications provides dynamic negotiation mechanism. More importantly, since users from different types of cloud computing have their own concerns, they need a different implementation structure of SLAs to integrate their own business rules to generate the target SLA suitable for them.

To compensate these shortcomings, some researchers introduced a conceptual framework defining several groups of

SLA parameters according to different types of cloud services [5]. However, there are still differences in the concerns about these parameters between users and providers. And some parameters are so hard to be quantified that they cannot calculate directly for evaluation. [6] suggested a SLA-based trust model for monitoring the execution of SLAs, by which cloud services can be estimated. The author did not, however, give the implementation and how to estimate the trust level.

In this paper, we optimize the division of SLA parameters and introduce a specific method to evaluate SLAs, which takes fair into account at the same time. Via this approach, it is convenient for users to select a more suitable cloud service and for providers to adjust their resource allocation strategy to avoid violations.

3. SLA Evaluation Model for Cloud Computing

Cloud services are usually be divided into four types: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). It is more specific that creating SLA agreement considering these four types of cloud services. But within a single agreement, what providers and users care about is not the same. The former cares more about the technical parameters that they can offer, such as functions and performance metrics. While the latter cares user experience and billing terms. This paper makes a further division to the service type of cloud computing, adding another dimension regarding providers and users.

A. General SLA Parameters for Cloud Services

In cloud computing, each SLA agreement has some basic parameters, such as security, billing, availability, reliability, et al. These parameters are agreed by both providers and users to confirm the responsibility and obligations of each party, as depicted in Table 1.

Table 1. General SLA Parameters for cloud services

	Parameter	Description
Provider SLAs	Throughput	The amount of data can be received within per time unit
	Security	Encryption, authentication and authorization
	Billing	Cost of service and method of calculating
	Monitoring	Monitoring agencies and monitoring method
	Privacy	The way of Storage and transmission
	Compensation	measures for SLA violations
	Load balance	Ability to handle concurrent access
	Recovery	Backup and ability to recovery from a disaster
User SLAs	Availability	Time proportion of Continuous delivery
	Reliability	Probability to keep providing specific service
	Response time	Time from request to receive a response
	Service level	Different service editions
	internationalization policy	Multi-language support
	Migration	Service migration to other platform
Support service	Service of help and support	

B. SLA Parameters for IaaS

Infrastructure as a Service mainly provides virtualization of computing resources, storage resources and network resources. Table 2 shows the key SLA parameters for IaaS. Within this type of SLA agreement, provider SLAs represent the performance metrics, while user SLAs are relating to user experience.

Table 2. SLA Parameters for IaaS

	Parameter	Description
Provider SLAs	CPU capacity	CPU speed of VM
	Memory capacity	Memory size of VM
	Booting time	Total time for boot a VM
	Storage	Size of the external storage of a VM
	VM scale	Max/min amount of VMs that can be used
	VM scheduling time	Time to increase/decrease a VM
	Auto scaling	Support auto-scaling or not
	Virtualization degree	Maximum number of VMs can be deployed on a physical host
	Bandwidth	Data transmission rate
User SLAs	Manageability	Time proportion of Continuous delivery
	Value-added service	Service of help and support

C. SLA Parameters for PaaS

Platform as a Service provides a series of tools covering the whole process of software engineering ranging from development to publishing, as well as the support of class libraries. The following parameters list in Table 3 shows what providers need to guarantee most, such as the ability of integration and scalability, and what users concerns more, such as the difference between developing on cloud and on real PC

Table 3. SLA Parameters for PaaS

	Parameter	Description
Provider SLAs	Integration	Ability to integrate services from other platform
	Scalability	Support Large-scale online users
	Network	Upload and download bandwidth offered to developer
	Application servers	Virtual resources that can be allocated to user's application
	Browsers	Supporting browsers for accessing PaaS
User SLAs	Development environment	Development tools and environment offered to user
	Deployment environment	Specific method for deployment and access

D. SLA Parameters for SaaS

Software as a Service provides an ordinary application based on Internet, which is very common in cloud service. From the perspective of providers, they need to offer functions like multi-terminal support and customization. From the perspective of users, they can consider usability. The specific SLA parameters are depicted in Table 4.

Table 4. SLA Parameters for SaaS

	Parameter	Description
Provider SLAs	Multi-terminal support	Support running on PC and mobile devices
	Open API	Provide API to individual developers
	VIP service	Additional services for VIP
	Offline service	Support offline operation
User SLAs	Usability	Easy to learn to use
	User-friendly	Attractive UI, abundant prompt
	Customization	Adapt to user's specific needs

As shown in the above 4 tables, the provider SLAs mainly focus on the functional requirements, while the user SLAs are mostly from the non-functional requirements. It is necessary to classify SLAs like that because providers have the advantage in monitoring and evaluating the system state. On the other hand, users are often confused with the technical part of SLAs, and they would rather to judge a service from their intuition. Considering the above problems, we believe that it is more practical to divide SLAs into provider SLAs and user SLAs, which also helping to provide a good understanding of the evaluation method in Section 4.

4. Evaluation Method based on the SLA Model

With numerous services in cloud environment, choosing a suitable cloud service merely relying on service types is not enough, especially for users. Therefore, users have to turn to the service agents for help, increasing the service cost.

The SLA model can not only determine the template parameters for providers and users who use different types of cloud service signing contract, but also helps users to select a superior service. The evaluation method based on the SLA model combines assessment from both providers and users. It is obvious that service with high grade has a high quality in its own service type. So, loss due to service violation will be greatly reduced in these services. In addition, service providers can also enhance their service level by improving the poor-graded services' resource scheduling strategy.

In order to understand this method, firstly, we introduced an evaluation vector $M(M_1, M_2, \dots, M_n)$, where M_i is the i th parameter. Both providers and users need to grade the evaluation vector, so as to generate a grade vector $P(P_1, P_2, \dots, P_n)$. Furthermore, each parties of the contract may put the emphasis on their own focus in the evaluation process of SLA. So there is a weight vector $W(W_1, W_2, \dots, W_n)$. Finally, the total quality of service Q can be calculated by applying the weighted vector W to the grade vector P . The entire evaluation model is illustrated in Figure 1.

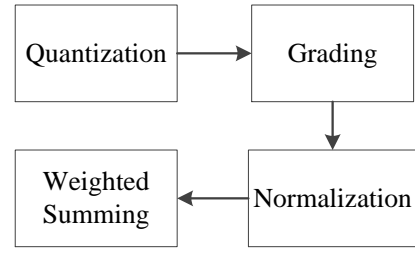


Fig. 1 The SLA Evaluation Model for Cloud Computing.

A. User SLAs Quantization

In Section 3, we divided SLA model into two parts: provider SLAs and user SLAs, representing functional requirements and non-functional requirements, such as CPU capacity and memory capacity. When evaluating this model, we also need to discuss both cases separately. For functional requirements, such as CPU capacity and memory capacity, SLOs are always be used as the evaluation indicator. However, non-functional requirements are mostly in the form of non-numerical, which cannot be directly calculated as a part of evaluation method.

To solve this problem, we present the following three methods to quantify user SLAs, which can turn the original evaluation vector $M(M_1, M_2, \dots, M_n)$ into a measurable evaluation vector $M'(M'_1, M'_2, \dots, M'_n)$, where M'_i can be the original M_i or a sub-vector..

1) Transformation

With this approach, user SLAs can be directly converted into provider SLAs which already have quantization metrics such as SLOs. E.g., the general SLA parameter *Service level* may have three values: Standard Edition, Professional Edition and Enterprise Edition. It can be correspondingly transformed into a providers SLAs set as $\langle Security, Billing, Recovery \rangle$, in which each SLA parameter has a SLOs to measure. Though the method is relatively simple, it is easy to introduce redundant user SLAs into the SLA contract. This approach is not recommended to overuse for it increasing the complexity objectively.

2) Inductive inference

Similarly, the inductive inference method will convert the user SLA into an n-tuple. Such as the general SLA parameter *Availability* can be concluded as a 2-tuple: $\langle A_{normal}, A_{failure} \rangle$. The elements are defined as follows:

$$A_{normal} = 1 - \frac{\sum_{i=1}^k (T_{end}^i - T_{start}^i)}{T} \quad (1)$$

$$A_{failure} = \frac{k}{T} \quad (2)$$

Where A_{normal} is the uptime ratio of service, $A_{failure}$ is service failure rate, T_{start}^i is the i th failure start time, T_{end}^i is the i th failure end time, and k is the total number of service failures in T period time. *Availability* is a compound concept,

not only including the ability of service to operating normally, also reflecting the stability (also known as low failure rate). So this parameter can be derived by comparing the target value and the actual value of the tuple. So can the other parameters, such as Usability, User-friendly and Manageable.

3) Statistical analysis

The statistical analysis is suitable for providers evaluating user SLAs. Providers, firstly, divide a user SLA into several grades with corresponding point. Then, invite a certain number of test users for trial and assessment. Finally, take the point of most-voted grade as the user SLA's metric. It can also take the average grade as the final metric depending on specific requirements.

B. Grading

In this phase, providers and users will apply SLOs and quantization metrics given above to produce a grade vector $P(P_1, P_2, \dots, P_n)$ and a weight vector $W(W_1, W_2, \dots, W_n)$, where each P_i is measured from M_i' , and W_i represents how important M_i is to evaluator. And W should also meet $\sum_{i=1}^n W_i = 1$.

C. Normalization

Due to each M_i' are different in dimensions, we use a normalization method to unify all P_i in the grade vector P into a scalar ranging [0,1], instead of summing up them directly. This is done by applying the following equation:

$$P_{i0} = \frac{P_i - lower_bound(M_i')}{upper_bound(M_i') - lower_bound(M_i')} \quad (3)$$

where P_{i0} represents the grade of normalized M_i' , $upper_bound(M_i')$ and $lower_bound(M_i')$ are the supremum and the infimum of M_i' . All P_{i0} can make up a new grade vector P_0 .

D. Weighted Summing

After the preparation of above three stages, we can now figure out the Quality of Service Q for single evaluator by apply the weight vector to the normalized grade vector P_0 as the following equation.

$$Q = \sum_{i=1}^n W_i \times P_{i0} \quad (4)$$

Considering both providers and users taking part in evaluation, their grade should also have different impact on the final result. In this paper, we suggest a weight ratio of 1:1 for this, deriving the following final Quality of Service equation:

$$Q_{total} = Q_{provider} + \frac{1}{n} \sum_{i=1}^n Q_{user_i} \quad (5)$$

where $Q_{provider}$ is the grade of service from provider, and the latter part of the equation is the average grade of service from users.

5. Conclusion

In this paper, we present an optimized SLA classification model for cloud computing and introduce a corresponding evaluation method. The model not only satisfies both cloud service providers and users requirements, but also makes the evaluation method more fair and reliable. In our future work, we will investigate to implement a simulation system of the proposed model. Furthermore, we will try to optimize and validate the SLA parameters and weighted evaluation equation.

6. Acknowledgment

We would like to thank to Mr. Wu Chenkang and Mr. Pan Shunhong for their help in model design. Our special gratitude must be paid to Prof. Zhu Yonghua for his guidance on writing. We also like to thank the Shanghai Educational Committee for their Scientific Researching and Innovating Fund support.

7. References

- [1] A. Sahai, S. Graupner, V. Machiraju, and A. Moorsel, "Specifying and Monitoring Guarantees in Commercial Grids through SLA," Proc. Third IEEE/ACM Int'l Symp. Cluster Computing and the Grid (CCGrid '03), pp.292-300, 2003.
- [2] Armbrust M., Fox A., Griffith R. "Above the Clouds: A Berkeley View of Cloud Computing," Berkeley, USA: University of California at Berkeley, Tech. Rep., 2009.
- [3] Keller A, Ludwig H. "The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services," Journal of Network and Systems Management, vol. 11, no. 1, pp. 57-81, 2003.
- [4] Andrieux, A., Czajkowski K., Keahey K., "Web services agreement specification (WS-Agreement)," 2004.
- [5] Mohammed A., Tharam D., Elizabeth C., "Conceptual SLA Framework for Cloud Computing," 4th IEEE International Conference on Digital Ecosystems and Technologies, pp. 606-610, 2010.
- [6] Mohammed A., Tharam D., Elizabeth C., "SLA-Based Trust Model for Cloud Computing," 13th IEEE International Conference on Network-Based Information Systems, pp. 321-324, 2010.