

HA-SLA: A Hierarchical Autonomic SLA Model for SLA Monitoring in Cloud Computing

Ahmad Mosallanejad, Rodziah Atan

Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Malaysia.
Email: ahmad.upm@gmail.com, rodziah@fsktm.upm.edu.my

Received 2013

ABSTRACT

Service Level Agreement (SLA) is a fundamental contract between service consumer and service provider which defined the qualities of agreed service. After SLA contraction, it should be monitored during the service invocations to confirm the service level objectives. Most of SLA structure and monitoring frameworks are brought from SOA and grid computing to cloud computing environment while they have different requirements. In this paper, a model is proposed as hierarchical autonomic (HA)-SLA based on cloud computing nature. It is proposed in this model that each SLA has connection with dependent SLAs in different layers of cloud computing, hierarchically, whereby each SLA should be able to monitor its attributes on its own. It is expected that HA-SLA model should able to increase SLA validity and users' confidence without compromising the respond time. The results produced by this model should be used as one of cloud computing quality of service assurance.

Keywords: Service Level Agreement (SLA); Cloud Computing; SLA Monitoring; Self-Management SLA

1. Introduction

Nowadays, interaction for services between consumers and providers are increasing dramatically within the cloud computing environment. A service level agreement (SLA) is a basis of all relations between consumers and providers. SLA includes service functionalities and quality of service (QoS) which identifies the service level objective (SLO). The SLA contents are defined after a negotiation process between service consumer and provider. Then, the user expects to use the service based on SLA features and the provider is responsible about any violated SLA. Therefore, an effective SLA monitoring system is essential to inspecting the SLO during the service invocations.

The current SLA and monitoring frameworks are not completely based on cloud computing requirements [1]. Most of them are designed for service oriented architecture (SOA) and grid computing which are having a different nature with cloud environment [2]. Some of these SLA monitoring systems are being applied in cloud computing without modification. Cloud computing architecture is hierarchical in nature. It includes the layers of Infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS).

This paper proposes a hierarchical autonomic (HA)-SLA model for SLA monitoring in cloud computing which has less attempt by the community. This model

provides a hierarchical SLAs based on hierarchical nature of cloud computing. So, each SLA is connected to the related SLAs in IaaS, PaaS and SaaS layers for sending an essential notification as a current SLA state. Moreover, each SLA is autonomic to manage and monitor its features itself which provides a self-management SLA. The proposed model is simulated and it is validated by experimental methodology.

In this paper, related works are discussed in Section 2, and HA-SLA model is described in Section 3. Next, the proposed model implementation is presented in Section 4. Finally, the HA-SLA evaluation and conclusion are illustrated in Sections 5 and 6 respectively.

2. Related Work

Many studies discussed SLA and SLA monitoring but less focus has been put in cloud computing area. In some of these researches, the SLA structure and monitoring system are brought from SOA and grid computing to cloud computing environment without enough attention to cloud computing unique features. This shortcoming is repeated in RESTful [3], SALmonADA [4] and SLA@SOI [5] frameworks.

An appropriate set of SLA attributes are described in [6] for different layers of cloud computing, separately and specified based on IaaS, PaaS and SaaS requirements; however, the hierarchical relation between dependent

SLAs is not considered in this research. Moreover, most of the related works such as [7-9] followed the common SLA structure from other environment which are not fulfilling the cloud computing features and nature.

Most of the related works described a centralized SLA monitoring system either in consumer, provider or trusted party sides. These models have their own weaknesses. A consumer side monitoring system is proposed in [10]. It does not have an access to provider resources for SLA attributes assessment. On the other hand, proposed framework in [11] is not completely trustful for consumer because it was located in the provider side. Trusted party in [12] also had a redundancy challenge because of its tracking mechanism.

Proposed LAYSI framework in [2] and another proposed model by [13] are the rare SLA monitoring systems which discuss hierarchical structure. The LAYSI distributes the monitoring process and SLA management through different actors of cloud computing and are not distributed through the different SLAs. The self-adaptive hierarchical monitoring system is proposed in [13] however also followed the common SLA structure.

According to the literature review, current cloud computing environment is lacking of effective SLA structure and SLA monitoring mechanism.

3. HA-SLA Model

In this paper, the HA-SLA model is proposed to portray SLA structure and monitoring mechanism based on cloud computing hierarchical nature. In this model, each SLA could assess its attributes itself as a self-monitoring SLA. The monitoring function is allocated in each SLA and it will be run per each service invocation to assess the current state of SLA. Moreover, each SLA is connected with related SLAs in different layers of cloud computing hierarchically as illustrated in **Figure 1**. Therefore, SLAs are able to notice their monitoring results to related SLAs as a current SLA state. The HA-SLA needs an infrastructure to work properly in hierarchical and autonomic manner. These features are described in subsections below.

3.1. Autonomic Feature

The HA-SLA model has an autonomic feature to provide a self-management SLA. Current SLA includes SLA functionalities, attributes, metrics and SLO as a simple contract document. The HA-SLA adds some functionality in SLA structure to make it as active entity rather than passive document. So, HA-SLA is an independent self-management SLA with embedded functions. SLA monitoring process is one of these functions.

In HA-SLA each SLA is able to monitor its attributes itself. They just receive the current metrics value from data collector then they assess the attributes state and

compare with SLO. Finally, it records the current SLA state to the log database and notifies it to any other related SLA. The results could be used in resource management system and for any critical reactions against violated SLA.

3.2. Hierarchical Feature

Cloud computing has a hierarchical nature and the upper layer (UL) services relied to lower layer (LL) architecture to work properly. If any service in IaaS is failed, all upper dependent services will be failed and its SLAs will be violated. Therefore the SLAs also rely to each other, hierarchically. However, current SLA do not have any relation to other related SLAs in cloud computing. The HA-SLA builds an effective hierarchical connection between dependent SLAs in different layers of cloud computing consisting of IaaS, PaaS and SaaS as illustrated in **Figure 1**.

The HA-SLA enables SLAs to announce their current states hierarchically as depicted in **Figure 1**. The lower SLAs notice the current SLA attribute values to the related upper SLAs after assessment process. Finally, these notifications could be used for current SLA attribute assessing, SLA monitoring, violated SLA detecting and resource reacting to increase the cloud computing reliability.

3.3. HA-SLA Architecture

The HA-SLA cooperates with some other components in each service provider to build a comprehensive hierarchical self-management SLA. **Figure 2** has presented the HA-SLA architecture from the service provider side. UL and LL ports are the provider gates for connecting to the related upper and lower layers SLA. At first, all SLA features are built after negotiation process. Then, HA-SLA is executed and receives the resource metrics from data collector. It then records the assessed SLA attributes in monitoring log database. Moreover the notification state is sent to related upper layer SLAs and resource management system for relevant decisions.



Figure 1. The hierarchical SLAs.

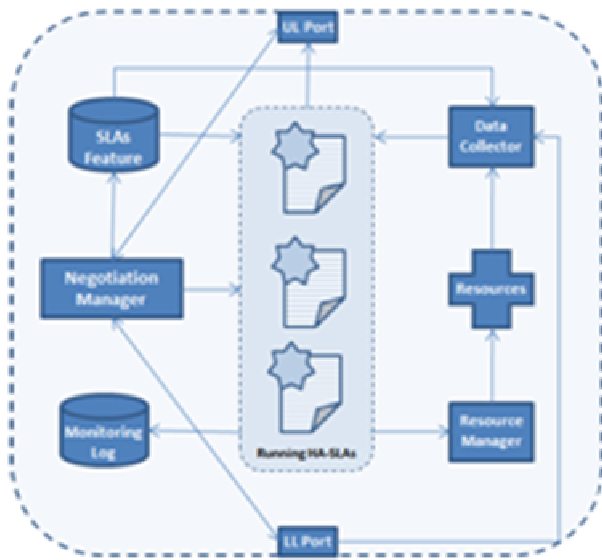


Figure 2. The HA-SLA architecture.

```

1.  Public struct Attribute
2.  {
3.      int code;
4.      string name;
5.      Metric[] metrics;
6.      string formula;
7.      float agreedV;
8.      char agreedOperation;
9.      float criticalV;
10.
11.     float assessor()
12.     {
13.         ...
14.     }
15. }
16. public struct HA_SLA
17. {
18.     int code;
19.     string name;
20.     Attribute[] attributes;
21.     int[] lowerSLAs;
22.     int[] emergency;
23.     int[] upperSLAs;
24.
25.     Notification[] Monitoring()
26.     {
27.         ...
28.     }
29. }
    
```

Figure 3: The HA-SLA struct.

4. HA-SLA Implementation

This research developed the HA-SLA model to prove the concept and functional workability. The HA-SLA structure includes SLA code, name, attributes, lower SLAs, upper SLAs and emergency services as presented in **Figure 3**. Moreover, monitoring function is located as a part of HA-SLA. The monitoring function invokes the attribute assessor to calculate the current value of each SLA attributes, and gets them compared to the agreed SLO. The attributes' value and SLA state are then recorded in monitoring log database, and notified to other related SLAs in upper cloud computer layer.

5. HA-SLA Validation

An experiment has been done to validate the HA-SLA functionality. A service throughput and response time attributes are collected from existing dataset in [14] which it is also used in [15,16]. These attributes are captured during the 339 invocations of the real web services. These data are the input of developed HA-SLA as a data collector output. The HA-SLA monitoring assesses the attributes value based on both initial data and notified lower layer SLA attributes value. This calculation is according to determined metrics and formula as illustrated in line 5 and line 6 of **Figure 3**. The HA-SLA monitoring results are presented in **Figure 4** and **Figure 5**.

The HA-SLA monitoring output for response time and throughput attributes are depicted in **Figures 4** and **5** respectively. **Figure 4** has illustrated the service response time value during the 339 invocations based on HA-SLA monitoring system. Moreover violated SLAs which they are more than agreed response time, 1 millisecond (ms), are detected. **Figure 5** has also presented these features

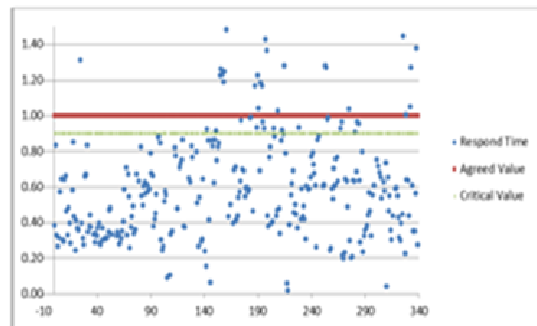


Figure 4: The HA-SLA monitoring results for response time.

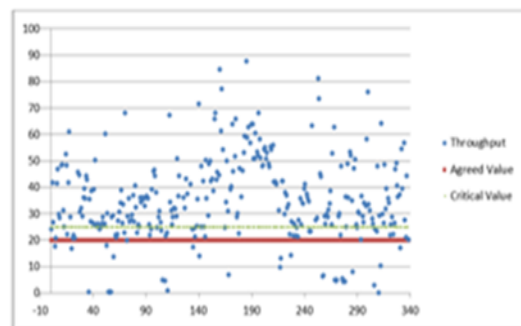


Figure 5. The HA-SLA monitoring results for throughput.

for throughputs which should not be less than 20 based on agreed SLO. Therefore the SLA attributes states are presented by developed HA-SLA monitoring clearly.

6. Conclusions

According to the literature review, the current SLA

monitoring systems have not completely covered the cloud computing requirements. Most of the SLA structure and monitoring systems were brought from SOA and grid computing without enough consideration to cloud environment hierarchical feature. In this paper, the HA-SLA model is proposed as a hierarchical self-management SLA. Each SLA has connected to the related SLAs in upper and lower layer of cloud computing hierarchically. Moreover, each SLA could monitor its attributes value itself and notify the current SLA state to the dependent SLAs. The proposed model is developed and the experiment validates the SLA structure and its monitoring functionality. The effective HA-SLA reacting functionality against violated SLAs will be considered in future works.

7. Acknowledgements

This project is funded by the Exploratory Research Grant Scheme, Ministry of Higher Education of Malaysia. Project no.: ERGS/1/2012/TK06/UPM/02/46.

REFERENCES

- [1] I. Ul Haq, I. Brandic, and E. Schikuta, "Sla validation in layered cloud infrastructures," *Economics of Grids, Clouds, Systems, and Services*, pp. 153-164, 2010.
- [2] I. Brandic, V. C. Emeakaroha, M. Maurer, S. Dustdar, S. Acs, A. Kertesz, and G. Kecskemeti, "LAYS: A Layered Approach for SLA-Violation Propagation in Self-manageable Cloud Infrastructures," in *Computer Software and Applications Conference Workshops*, 2010, pp. 365-370.
- [3] R. Kübert, G. Katsaros, and T. Wang, "A RESTful implementation of the WS-Agreement specification," in *Proceedings of the Second International Workshop on RESTful Design*, 2011, pp. 67-72.
- [4] C. Muller, M. Oriol, M. Rodríguez, X. Franch, J. Marco, M. Resinas, and A. Ruiz-Cortes, "SALMonADA: A platform for monitoring and explaining violations of WS-agreement-compliant documents," in *Principles of Engineering Service Oriented Systems (PESOS), 2012 ICSE Workshop on*, 2012, pp. 43-49.
- [5] M. Comuzzi, C. Kotsokalis, G. Spanoudakis, and R. Yahyapour, "Establishing and monitoring SLAs in complex service based systems," in *Web Services, 2009. ICWS 2009. IEEE International Conference on*, 2009, pp. 783-790.
- [6] M. Alhamad, T. Dillon, and E. Chang, "Conceptual SLA framework for cloud computing," in *Digital Ecosystems and Technologies (DEST), 2010 4th IEEE International Conference on*, 2010, pp. 606-610.
- [7] P. Khanna and B. Babu, "Cloud Computing Brokering Service: A Trust Framework," in *CLOUD COMPUTING 2012, The Third International Conference on Cloud Computing, GRIDs, and Virtualization*, 2012, pp. 206-212.
- [8] H. P. Borges, J. N. de Souza, B. Schulze, and A. R. Mury, "Automatic generation of platforms in cloud computing," in *Network Operations and Management Symposium (NOMS), 2012 IEEE*, 2012, pp. 1311-1318.
- [9] A. V. Dastjerdi, S. G. H. Tabatabaei, and R. Buyya, "A dependency - aware ontology - based approach for deploying service level agreement monitoring services in Cloud," *Software: Practice and Experience*, vol. 42, pp. 501-518, 2011.
- [10] B. S. ARAB and A. A. A. B. D. GHANI, "EMPLOYING PERFORMANCE COUNTERS AND SOFTWARE WRAPPER FOR MEASURING QOS ATTRIBUTES OF WEB SERVICES," *Journal of Theoretical and Applied Information Technology*, vol. 34, 2011.
- [11] E. Badidi, "A framework for brokered Service Level agreements in SOA environments," in *Next Generation Web Services Practices (NWeSP), 2011 7th International Conference on*, 2011, pp. 37-42.
- [12] R. Jurca, B. Faltings, and W. Binder, "Reliable QoS monitoring based on client feedback," in *Proceedings of the 16th international conference on World Wide Web*, 2007, pp. 1003-1012.
- [13] G. Katsaros, G. Kousiouris, S. V. Gogouvitis, D. Kyriazis, A. Menychtas, and T. Varvarigou, "A Self-adaptive hierarchical monitoring mechanism for Clouds," *Journal of Systems and Software*, 2011.
- [14] Z. Zheng. *Distributed Reliability Assessment Mechanism for Web Services*. Available: <http://www.wsdream.net/>
- [15] Y. Zhang, Z. Zheng, and M. R. Lyu, "Exploring latent features for memory-based QoS prediction in cloud computing," in *Reliable Distributed Systems (SRDS), 2011 30th IEEE Symposium on*, 2011, pp. 1-10.
- [16] Z. Zheng, Y. Zhang, and M. R. Lyu, "Distributed qos evaluation for real-world web services," in *Web Services (ICWS), 2010 IEEE International Conference on*, 2010, pp. 83-90.