SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

VOLUME 3, ISSUE 1
ISSN 2054-7390

Development of SLA Monitoring Tools Based on Proposed DMI in Cloud Computing

[1]Nor Shahida Mohd Jamail, [2]Rodziah Atan, [3]Rusli Abdullah and [4]Mar Yah Said
[1,2,3,4] Department of Computer Science and Information Technology, University Putra Malaysia, Msia;
[1]Department of Art, Computing and Creative Industry, Sultan Idris Education University, Msia;
[1]shahida@fskik.upsi.edu.my; [2]rodziah@upm.edu.my; [3]rusli@upm.edu.my; [4]maryah@upm.edu.my

ABSTRACT

Service level agreement (SLA) is a contract between service provider and user about the quality of service (QoS) in cloud computing. The cost value and benefit value of SLA monitoring systems is a concerned issue in cloud computing. The trustable SLA monitoring model is important to assess SLA validation in cloud computing. The optimization of monitoring interval is another objective of study to economize SLA monitoring system. Therefore in this paper the SLA monitoring tools is developed to evaluate the proposed dynamic interval in actual test bed of cloud computing. The experiment design described in this paper presents the cloud configuration and environmental setup for test bed experiment.

Keywords: Service Level Agreement; Dynamic Monitoring Interval; Cloud Computing; Cost; Monitoring Tools; Quality of Service.

1 Introduction

Service level agreement (SLA) is a contract between service provider and user upon quality of service (QoS) in cloud computing [1]. SLA contains the agreed attributes and the value of service level objectives (SLO). Service provider should deliver services based on agreed quality in SLA. If quality value of running service exceeds the agreed SLO, service provider must pay penalty for this SLA violation. The SLA monitoring tools is unavoidable to evaluate agreed SLAs at the run time and detect any probable SLA violations [2]. Both service provider and consumer need to monitor the QoS to be sure about SLA validity. The cost value and benefit value of SLA monitoring systems are a concerned issue in cloud computing [3]. The SLA monitoring systems need resources consumption consisting CPU, Memory, and Storage for execution. The amount of consumed resources by monitoring system is the cost of SLA evaluation. On the other hand, SLA monitoring systems make benefits by detecting the SLA violations because service provider afterward can adapt the infrastructure to prevent more numbers of SLA violations and avoid penalty cost. This study proposes the SLA evaluation model to increase the adaptability of SLA monitoring systems and subsequently detect the most probable SLA violations with a reasonable overhead cost. The costs and benefits of SLA monitoring system is the main focus of this study. Interval value of SLA monitoring system is the central concentrated issue in this area because it has a high impact on cost value and benefit value of monitoring system. An existing cloud service is executed in developed test bed and the predefined SLA attributes are evaluated by developed monitoring tools.

DOI: 10.14738/tmlai.31.841
Publication Date: 24th January, 2015
URL: http://dx.doi.org/10.14738/tmlai.31.841

2 The Proposed Adaptive SLA Evaluation Model

The trustable SLA monitoring model is proposed to assess SLA validation in cloud computing [4]. The optimization of monitoring interval is another objective of study to economize SLA monitoring system. The SLA monitoring tools is developed to evaluate the proposed dynamic interval in actual test bed of cloud computing. The proposed trustable SLA monitoring model is presented Figure 1. Proposed model contains three actors including Service Provider (SP), Trusted Party (TP), and User. The negotiation process between SP and user is done by TP [5] cooperation before SLA monitoring process. The agreed SLA, then, is recorded in TP database for run time monitoring activities. The monitoring engine is installed in SP center by TP to collect the raw data for assessing the agreed SLA attributes. Although the monitoring engine is located in SP center but SP does not have permission to access or manipulate the collected data by monitoring engine. Monitoring engine, indeed, is a part of trusted party but located in SP center for data collection purposes.

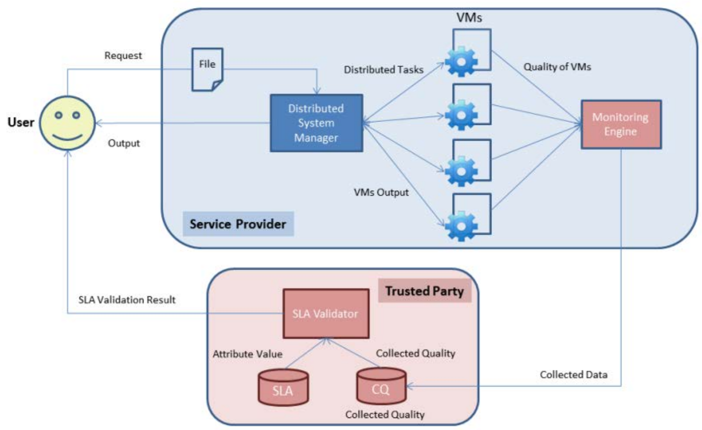
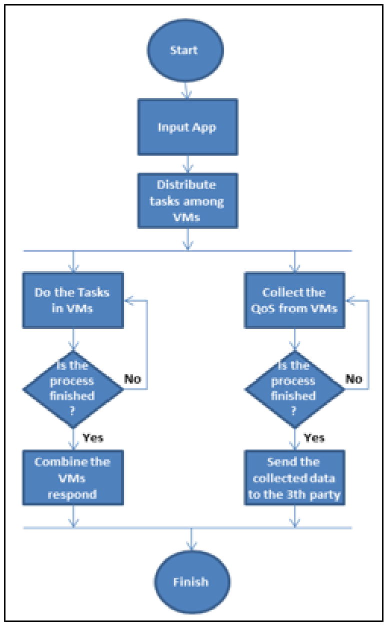


Figure 1: Proposed adaptive SLA evaluation model

The Distributed System Manager (DSM) is located in SP center to manage the task scheduling and process. DSM gets the user request which contains the name of agreed service and any probable input data and files. DSM checks the request validity and employs the needed VMs based on agreed SLA. The task is then distributed among VMs to execute the process. Each VM responds the output of the process to DSM. Finally, DSM combines the received results from VMs and sends the output to the user. Monitoring engine collects the value of SLA attributes from VMs when the tasks are processing. The collected data send to the TP center for SLA evaluation. SLA validator maps the collected data to the SLA attributes and then compares the actual quality to the agreed SLO. The SLA validation result is sent to the user to be aware about SLA validity and SLA violations. **Error! Reference source not found.** presents the activity diagram of service running and SLA monitoring. The tasks are distributed among VMs after receiving the user request and input file. The task processing and QoS collecting are done at the same time. The respond results from VMs are combined by DSM and the collected data about QoS are sent to the TP at the same time. Finally, user gets the output of service from SP and also the SLA validation report from TP.



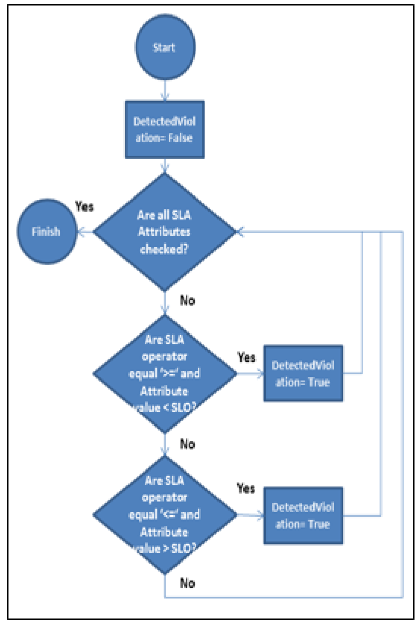


Figure 2: Activity-diagram of SLA **Figure 3: SLA Evaluation Process**

Figure 3 shows the SLA evaluation process which is done by TP. In the first step, process assumes that the SLA is not violated as a default. The collected data from VMs are mapped from low-level metrics to high-level attribute based on defined formula in SLA. The measured value of attribute then compares with agreed SLO based on defined operator for that attribute in SLA. If the attribute value exceeds the agreed SLO, the SLA is violated. This process is repeated for all stated attributes in SLA.

3 SLA Monitoring Tools Based on Proposed DMI

The dynamic monitoring interval (DMI) is proposed to economize the SLA monitoring system. DMI aims to reduce the overhead of monitoring system when the agreed service is working normally. Moreover, The SLA violations should be detected when the quality of agreed service exceed the defined quality in SLA. The number of missed violation detection should be minimized. The following DMI formula is proposed to adapt the interval value based on the cost and benefit values of monitoring system. Monitoring Cost (MC) is the sum of needed Storage, Memory and Processor resources which are consumed by monitoring system in execution time. The cost of each resource measures by multiplying the amount of resource usage in the cost unit of the resource. The main objective of SLA monitoring system is the SLA violation detection to prevent further violation by suitable reactions. It reduces the total penalty cost which should be pained to the user. The Violation Detection Benefit (VDB) is the benefit of monitoring system which it is measured by multiplying the penalty cost in the number of detected violations.

$$Monitoring\ Interval\ (I) = \begin{cases} \frac{I}{2}, & MC < VDB \\ I\,, & MC = VDB \\ I + \frac{I}{4}, & MC > VDB \end{cases}$$

$$Monitoring\ Cost\ (MC) = \sum_{i=0}^{n} (S * SCU + M * MCU + P * PCU)$$

$$Violation\ Detection\ Benefit\ (VDB) = \mathrm{PC} * \sum_{i=0}^{n} DV$$

S: Storage usage
SCU: Storage cost unit
M: Memory usage
MCU: Memory cost unit
P: Processor usage
PCU: Processor cost unit
PC: Penalty cost
DV: Detected Violation
n: number of monitoring iterations

The monitoring interval becomes the half of interval value when the VDB is larger than MC. The agreed service is running without or with a few violations in this period so the monitoring interval is deducted to reduce the monitoring cost. On the other hand, the quarter of monitoring interval is added to the interval value when the MC is larger than VDB. In this period, significant number of SLA violations is detected and the monitoring system should run faster to increase the profit of monitoring system and detect any probable SLA violations. The interval value is not changed if the MC and VDB are equal. A SLA monitoring tools is developed based on proposed DMI to monitor the agreed SLAs. Figure 4 shows the developed SLA monitoring tools at the run time. It is monitoring the processor, memory and storage usage value of employed VMs. Each resource usage is presented in forms of total consumption and VMs consumption. Detected SLA violations and the violated attribute value are shown in a column. The cost and benefit of monitoring tools are also measured at the run time and they are used in monitoring interval adaptation process. The changes of interval value are presented in Interval TextBox.

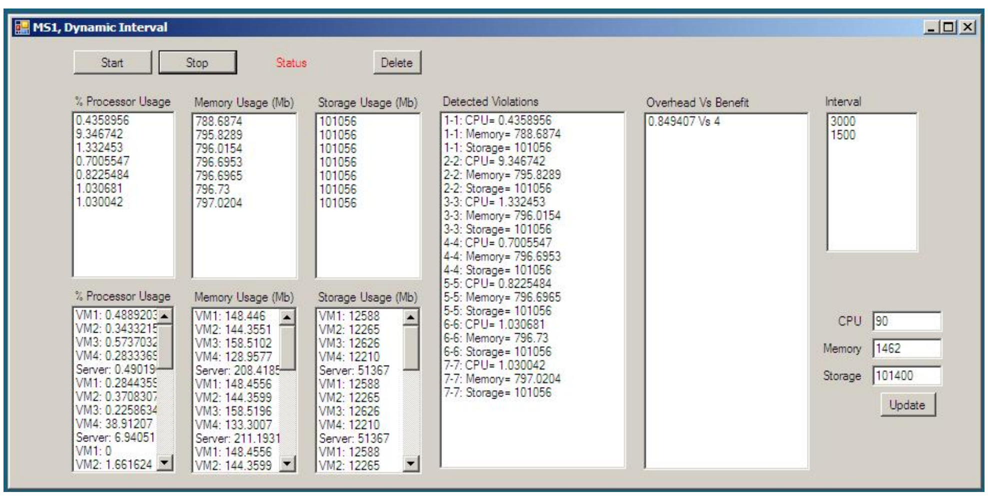


Figure 4: Snapshot of developed SLA monitoring tools

Figure 4 presents the monitoring results of agreed SLA that the CPU, Memory and Storage usage should be more than 90%, 146 MB, and 101400 MB respectively. The resources of 4 VMs are monitored periodically and the total CPU, Memory and Storage usage are also presented. The overhead and benefit of monitoring tools are measured every 5 iteration of SLA evaluation. The monitoring interval value is adopted based on measured cost value, benefit value and proposed DMI formula. The benefit of monitoring tools is 4 $ versus 0.84 $ cost in first assessment; and the monitoring interval is changing from 3000 MS to 1500 MS.

4 Test bed Experiment

The test bed is developed to provide the experiment environment for running cloud services and executing the developed monitoring tools. Three physical machines are employed producing 4 VMs and 1 Vcenter server as presented Figure 5 Vmware vSpher 5.1 is installed on the server to configure and manage VMs. ESXi 5 is installed on two physical machines as a hypervisor to create the needed VMs. 4 VMs are created and they are managed remotely by Vmware vSpher application from server.

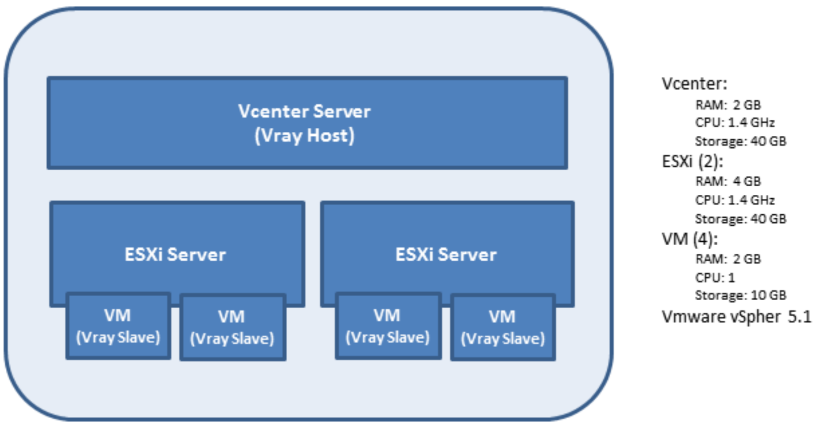


Figure 5: Test bed specifications

V-Ray Host application is installed on the server to do the image rendering process by employing the installed V-Ray Slaves on VMs. V-Ray Host takes the 3dsMax file and distributes the different parts of image to the VMs for rendering process. Each V-Ray Slave renders the assigned image parts and responds the output to the V-Ray Host for finalization of complete rendered image. The developed SLA monitoring system is installed on the server to monitor the VMs. The physical machine server has 2 GB RAM, 1.4 GHz CPU, and 40 GB storage capacities. Each hypervisor contains 4 GB RAM, 1.4 GHz CPU, and 40 GB storage properties which assigns 2 GB RAM and 10 GB storage capacity to each virtual machine.

5 Performance Evaluation

Certain development tools are used to implement the test bed, cloud service scenario, and monitoring tools. The virtualized environment is created by employing the Vmware productions consisting ESXi 5.0 and Vmware vSpher 5.1. ESXi is a hypervisor to manage and assign the resources to the created VMs [6]. Vmware vSpher is employed as a remote application to create and configure the VMs on ESXi. The SLA monitoring tools is developed using Visual Studio .Net. Standard tools and libraries of .NET Framework 3.5 among the functions used, such as System. Threading and System. Diagnostics. Performance. Counter for data collection. Autodesk 3ds Max 2012 and V-Ray Host 2.0 plug-in are installed on the server for running the distributed image rendering on experimental case. The Slave version of V-Ray is installed on each VM for V-Ray Host to delegate tasks. The value of SLA attributes are measured by SLA monitoring tools and the outputs are collected per each employed monitoring interval based on certain criteria setting. The results are presented in forms of scatter graphs for predefined situations. The cost of SLA monitoring tools are calculated based on consumed CPU, Memory, and storage resources for SLA monitoring execution. A formula is defined and follows for cost calculation. The amount of each resource usage is multiply on the predefined price value per unit to assess the spend cost. The cost assessment formula is:

$$C = St * SCU + Me * MCU + CP * CCU$$

While C refers to the monitoring cost. St, Me, and CP are storage, memory and CPU resources respectively. SCU, MCU, and CCU are the cost units for storage, memory and CPU accordingly. Table 1 presents the predefined price value per unit for each resource usage.

Table 1: Predefined price value per unit

Resource	Price per Unit
Storage	0.04 $ per 1 MB
Memory	0.15 $ per 1 MB
CPU	0.15 $ per 1% Usage

The benefit (B) of SLA monitoring tools are measured based on the number of detected violations (DV) multiplying on the predefined penalty value (PC) in SLA as stated in following formula.

$$\mathrm{B} = \mathrm{PC} * \sum_{i=0}^{n} DV$$

The predefined penalty value is 0.8 $ per SLA violation in all situations. This penalty value is inspired from predefined penalty range of SLA violations [7]. The cost and benefit of SLA monitoring tools is measured at the run time by employing an intrusion code in the test bed [8]. The descriptive statistics is employed for measuring and presenting the cost and benefit of monitoring tools in different situations. The total profit (Pr) of each monitoring tools execution is assessed by deducting the measured cost value from benefit value.

$$Pr = C - B$$

6 Conclusion

The trustable SLA monitoring model is proposed in this chapter. The actors of proposed model included the end user, service provider (SP), and trusted party (TP). The agreed SLAs are recorded in (TP) center and the monitoring engine of TP is located in SP to collect the needed data about agreed QoS. Monitoring engine sent the collected data to the TP center after executing the tasks by employed VMs. The SLA evaluation results and probable detected violations are finally reported to the user. The proposed dynamic monitoring interval (DMI) is also described in this paper. The interval value [9] is adopted based on the cost and benefit assessments of monitoring system. The DMI increased the monitoring interval when the benefit value of monitoring system is higher than the cost value. The monitoring system, on the other hand, decreased the interval value when the measured cost of monitoring system is more than its benefits. Finally a SLA monitoring tools is developed based on proposed DMI to monitor the agreed SLA and adopt the monitoring interval [10] at run time. The experiment test bed and virtualized environment are made ready to prove the model execution. The distributed image rendering service is executed as example cloud service in the test bed, and monitored by the developed monitoring tool. The predefined SLA contains specific SLO values gained from strict reviews of related works. The cost and benefit values of SLA monitoring tools are measured at the run time using the defined formula.

REFERENCES

[1]. Patel, P., Ranabahu, A. H., and Sheth, A. P., *Service level agreement in cloud computing*. OOPSLA Cloud Computing workshop, 2009. p. 1-10.

[2]. Boniface, M., et al., *Dynamic service provisioning using GRIA SLAs.* Service-Oriented Computing-ICSOC 2007 Workshops. 2009. p. 56-67.

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

[4]. Anithakumari, S., and Sekaran, K. C., *Autonomic SLA Management in Cloud Computing Services.* Recent Trends in Computer Networks and Distributed Systems Security. Springer, 2014. p. 151-159.

[5]. Barbosa, A. C., et al., *Evaluating architectures for independently auditing service level agreements.* Future Generation Computer Systems, 2006. Vol.22. No.7. p. 721-731.

[6]. Wood, T., et al., Sandpiper*: Black-box and gray-box resource management for virtual machines.* Computer Networks, 2009. Vol.53. No.17. p. 2923-2938.

[7]. Emeakaroha, V. C., et al., *Casvid: Application level monitoring for sla violation detection in clouds*. IEEE 36th Annual Conference on Computer Software and Applications (COMPSAC), 2012. p. 499-508.

[8]. Emeakaroha, V. C., et al., *Towards autonomic detection of sla violations in cloud infrastructures.* Future Generation Computer Systems, 2012. Vol.28. No.7. p. 1017-1029.

[9]. Maurer, M., et al., (2010). *Towards knowledge management in self-adaptable clouds*. 2010. 6th World Congress on Services (SERVICES-1), p. 527-534.

[10]. Maurer, M., Brandic, I., and Sakellariou, R., *Adaptive resource configuration for Cloud infrastructure management.* Future Generation Computer Systems. 2013. Vol.29. No.2. p. 472-487.