Scientific Research Publishing

Journal of Computer and Communications, 2020, 8, 78-90
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Cloud Services Quality Enhancement through Control & Monitoring of Service Level Agreements at Server and Consumer Side

Abel Adane

Faculty of Computing and Software Engineering, Institute of Technology, Arba Minch University, Arba Minch, Ethiopia
Email: abeladi6@gmail.com

How to cite this paper: Adane, A. (2020) Cloud Services Quality Enhancement through Control & Monitoring of Service Level Agreements at Server and Consumer Side. *Journal of Computer and Communications*, **8**, 78-90.
https://doi.org/10.4236/jcc.2020.88007

Received: June 28, 2020
Accepted: August 28, 2020
Published: August 31, 2020

Abstract

Several researches and case studies have been revealing a repetitive issue and that is monopolization of controlling, measuring, and metering of the consumer's billing for the consumed Cloud Services. The Cloud Service Providers (CSPs) have been hiding, or monopolizing the controlling, metering tools and ignoring the calculation and consideration of the service defects like downtime, service outage, poor performance and client service migration delays during overloads of the servers. The main issues behind these discrepancies are the Service Level Agreement (SLA), its violation, and the monopolized Cloud Governance. Till now there is a lack of parallel monitoring and metering system of the consumed cloud services at the customer level under the provisions of SLAs. The Cloud Governance Tier has not given any flexibility or utility to monitor such consumed cloud services at the client tier in parallel to the CSP tier. This research is an effort to develop a SLA controlling/monitoring framework to solve the identified problems other than the CSP level. Further the E_Draw max was used for designing the framework. The framework consists of the components like; SLA_CSP, SLA_CSC, SLA_M&C and Cloud Services and Delivery Models. After designing the framework, a prototype was developed using CloudSim on java net beans, and MySQL database. This prototype was implemented using four scenarios to evaluate its performance and the functionalities. 1) Checking the SLA under CSP whether the resources to the consumer in the same manner with promised. 2) To check if the request from cloud consumer is less than what is expected in SLA. 3) To check by increasing the request of cloud consumer more than what is expected in SLA and 4) Using the SLA_M&C mechanism. After execution it was observed that, in the first and the second cases there is no violation, but there is violation in the third and fourth cases. This implied that there are chances of the discrepancies in resources provided to make the selves beneficiary beyond

SLA under CSP and CSC was discriminated while receiving services from CSP.

Keywords

Cloud Computing, Service Consumer, Cloud Service Provider, SLA, SLA_CSC, SLA_CSP, SLA_M&C

1. Introduction

Nowadays information communication technology is widely spreading all over the world from time to time. This has a very great impact on the development of every country in this world. Communication is the back bone for all in the development of all the countries, economically, socially and politically. Information communication was used different model to deliver information for the consumer of services provided to the customers. The services rendered by the cloud computing can be accessed in anytime, anywhere over any device manners easily, user-friendly and conveniently. Several researches and case studies have been revealing a repetitive issue and that is monopolization of measuring, and metering of the consumer's billing for the consumed Cloud Services. Now a day the world migrates to the cloud computing technology to provide and consume the services provided on cloud. According to NIST [1] cloud computing defined as" Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction". There are essential characteristics of cloud computing *i.e.* on demand self-services, broad network access, resource pooling, Rapid elasticity and Measured service [2] [3] [4] [5] [6]. There are three basics service delivery model in cloud computing environment namely, software as a service, platform as a service (SaaS, PaaS and IaaS) respectively. One of the best characteristic of cloud computing is pay per use for services means we only par for the services we used. This is logically acceptable and what makes cloud computing spreading out worldwide. But some cloud providers are not always responsible for the services they promised to deliver for the end user or consumers over the cloud. The guarantee of both cloud service providers and cloud service consumers is service level agreement. The Cloud Service Providers (CSPs) have been hiding, or monopolizing the metering tools and ignoring the calculation and consideration of the service defects like downtime, service outage, poor performance and client service migration delays during overloads of the servers. The main issues behind these discrepancies are the Service Level Agreement (SLA), its violation, and the monopolized Cloud Governance. Till now there is a lack of parallel monitoring and metering system of the consumed cloud services at the customer level under the provisions of SLAs. The Cloud Governance Tier has not given any flexibility or utility to monitor such consumed cloud services at the client tier in parallel to the CSP tier. We do have three main components in cloud computing environment. The first layer of this component is cloud service consumers or brokers. The second layer is application like web, mobile, desktop or laptops which are used to request the services to the cloud services provider. The third layer includes three important sub components like service request examiner, SLA management & resource allocation and cloud service provider.

Figure 1 indicated that structure of the cloud computing architecture in the cloud computing paradigm. In this figure, there are 5 layers namely cloud service user, Application (web, mobile and or desktop), service request examiner (sre) and admin controller (AC), sla Management and cloud service provider/cloud resources.

2. Literature Review

The landscape of cloud computing has significantly changed over the last decade. Not only have more providers and service offerings crowded the space, but also cloud infrastructure that was traditionally limited to single provider data centers is now evolving [7]. Number of Researchers tried to do different researches on cloud service level agreement [8]. According to [1] the Cloud Computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the datacenters that provide those services. The services themselves have long been referred to as Software as a Service (SaaS), so we use that term. The datacenter hardware and software is what we will call a Cloud. When a Cloud is made available in a pay-as-you-go manner to the public, we call it a Public Cloud; the service being sold is Utility Computing. Current examples of public Utility Computing include AmazonWeb Services, Google AppEngine, and Microsoft Azure. We use the term Private Cloud to refer to internal datacenters of a business or other organization that are not made available to the public [9]. In cloud computing every service are depend on service level agreement that is used as a guarantee for both parties (cloud service consumers and cloud service providers). As mentioned in the abstract part of this paper a lot of problem from most of the cloud service providers were happened. In addition to that according to [10] different problems were happened because of the violation of service level agreement.

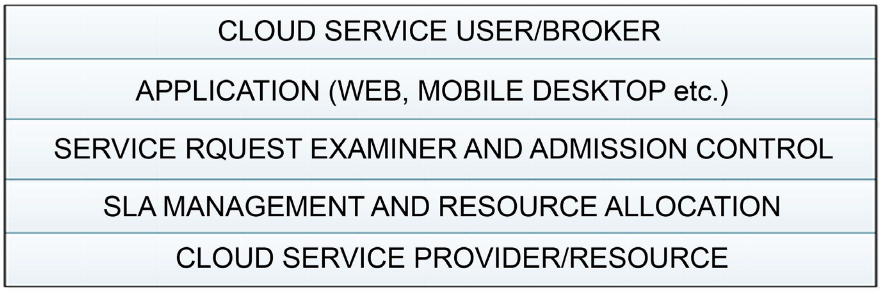
CLOUD SERVICE USER/BROKER
APPLICATION (WEB, MOBILE DESKTOP etc.)
SERVICE RQUEST EXAMINER AND ADMISSION CONTROL
SLA MANAGEMENT AND RESOURCE ALLOCATION
CLOUD SERVICE PROVIDER/RESOURCE

Figure 1. The basic structure of cloud computing architecture.

- Administration tools, systems, consoles and other related software restricted to only those with authorization.
- Access restricted to the minimum time and capability required to perform an authorized activity.
- System administrators restricted from accessing sensitive IT assets through the use of cryptographic, authentication and other techniques.
- Backup and log data protected through segregation of administrator duties and environments.

Cloud Computing is a huge scale distributed computing prototype that is manage by economies of scale, in which services are provided on demand over the internet for customers. Central remote servers and Internet are used to maintain application and data in cloud computing [10]. Cloud computing assign remote services with a user's data, computation and software. It is a model which provide on demand access to a shared pool computing resource like servers, storage, networks, applications and services. Cloud computing and storage solutions provide users and enterprises with various capabilities to store and process their data in third-party data centers. It depends on sharing of resources to acquire coherence and economy of scale over a network. Due to the advantages like cheap cost of services, high performance high computing power, scalability, accessibility and availability cloud computing has become demanded [11]. A study by Durga Prasad Sharma & Yemane tried to develop a client side monitoring system for greenness of cloud services however this research focuses on both client and server side monitoring and control for all the services rather than only green services [12].

Figure 2 shows the basic cloud service delivery models in cloud computing. They are software as a service (SaaS), plate form as a service (PaaS) and Infrastructure as a service (IaaS) respectively.

3. Research Methodology

Starting from the last two decades many researchers are performed number of researches on cloud computing technology and cloud based service level agreement. In almost all cloud computing researches related service level agreement,

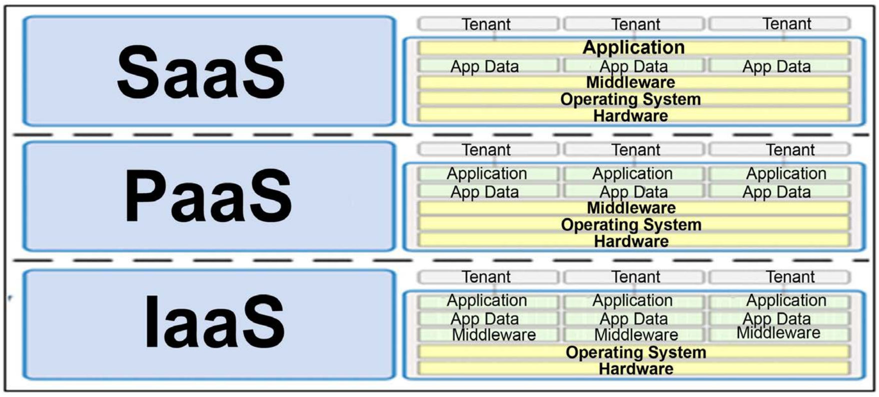

Figure 2. Cloud service delivery models.

the researchers were focused on cloud service provider side service level agreement. But monitoring service level agreement from cloud service provider side does not answer the questions of cloud consumers/clients. Clients need to ensure that, the cloud characteristic which is pay-per-use must be compromised. Because of different factors cloud consumers are paid beyond the services delivered and as a result the service level agreement was violated. When the service level agreement is violated most of the cloud providers hide the issue internally and the consumers had no information about how much the services are delivered. This is because of the service level agreement was monitored only from cloud services providers. So we considered this challenged problem, and developed the best framework which is used to monitor service level agreement from client side monitoring service level agreement. This doesn't mean that service level agreement is not monitored from cloud services providers' side. Since every cloud services providers have their own services level agreement, in this research we focused to add how to monitor service level agreement in cloud service consumers/client side. We put the details of our research under the title of research out comes.

As every research has its own research methodology we have used to develop our framework using the following methodology. We used the secondary data source to do this research which is collected using different literature reviews and different cloud providers' web site to well understand cloud environment as well as well identification of the problems in detail. After the problem is well identified we used the much related papers or research for further study. So since we completely check as our research was not done yet, we have developed our new framework in which the services are delivered in the control and measurement of the three parties' namely cloud provider, cloud consumer and Third party/components. Next to this we implemented to our framework and finally we have checked or evaluated our framework at the end.

Figure 3 indicated the way we followed during the research work step by step. We used the secondary data sources for our research development. Next to that we red different literature reviews and cloud web sites like amazon web services, we identified the problems related to the cloud service level agreement. After we identified the problems we developed the new framework, implemented the developed framework and evaluated it respectively.

Tool Selections

We had used different tools to complete this research. We used Microsoft office for different proposes like documentation proposes, E_Draw Max for developing framework and drawing different diagrams used in this research. We used Java Net beans for customizing cloudsidm tools to evaluate the developed framework. Cloudsim was developed using java programming language and as a result we used it to run the implementation of the framework on java net beans. Database used to store the results of delivered quality of services and service level agreements that guided the main actors of the cloud services was stored in database which is MySQL server.

Figure 4 shows the tools used in this research. Accordingly we used Microsoft office for documentation purposes, E_Draw Max drawing framework and necessary figures, MYSQL server to create database and Java net beans for programming purposes.

4. Framework Design & Discussion

Our research was focused on the cloud computing service level agreement and we found out a lot of problems in cloud based service level agreement while doing this research. We investigated that most of the cloud providers monitor and measure the cloud service level agreement just for the sake of their business. This result into serious problems for cloud service consumers. In our research we found that the best solution for the problems listed in the introduction part of this research document. The researchers understand the problem and want to develop new framework that reduce the problem mentioned above. If the service level agreement is violated cloud providers hides the parameters that was under the quality of services mentioned in the service level agreement, which leads to the service consumers un happy and discrimination will be happened.

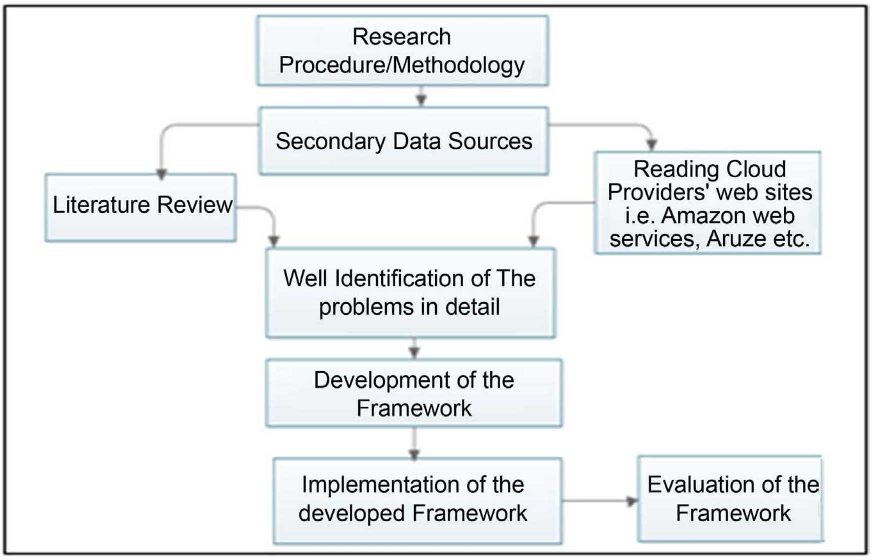
Research Procedure/Methodology
Secondary Data Sources
Literature Review
Reading Cloud Providers' web sites i.e. Amazon web services, Aruze etc.
Well Identification of The problems in detail
Development of the Framework
Implementation of the developed Framework
Evaluation of the Framework

Figure 3. Research Methodology of the research.

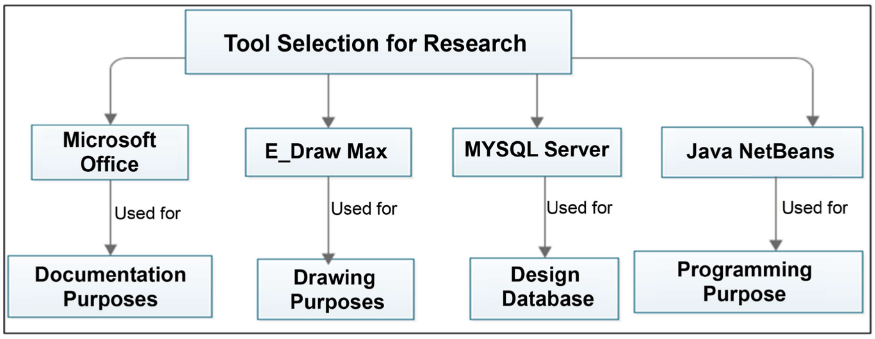

Figure 4. Tool selection for the research.

This service level agreement is controlled and monitored by cloud providers and consumers not monitored and controlled only by cloud service provider.

Figure 5 indicated the new framework we developed in this research. There are different components and this figure. Cloud service provider who deliver services over cloud, cloud service consumer who consumed the services over cloud, cloud services delivery models that is forms of the services, cloud resources, database, service level agreement components, components that are included like responsibility define service, objectives, actions and Guarantee and SLA M&C which is used to monitor and control the parameter mentioned above *i.e.* QoS measuring, cost of services, services measurement, service usage, down time and bandwidth.

In this framework developed from 8 components which are cloud service provider, cloud service consumer, cloud service delivery models like Iaas, paas and saas, cloud resources, Service level agreement, service deployment models like private, community, public and hybrid cloud, database and SLA M&C (Service level agreement monitoring and controlling) components. Cloud service provides is responsible to provide the services to the cloud service consumer according to the service level agreement signed between them. So the cloud resources and the service level agreement must be controlled by the cloud provider. The second component is cloud service consumer and is responsible to pay per use for the services they accessed. In this framework the service level agreement also must be monitored and controlled by the cloud service consumer. Because there are different problems when service level agreement was monitored and controlled only from cloud service provider side. For example the quality of services mentioned in the service level agreement and the delivered services are not matches

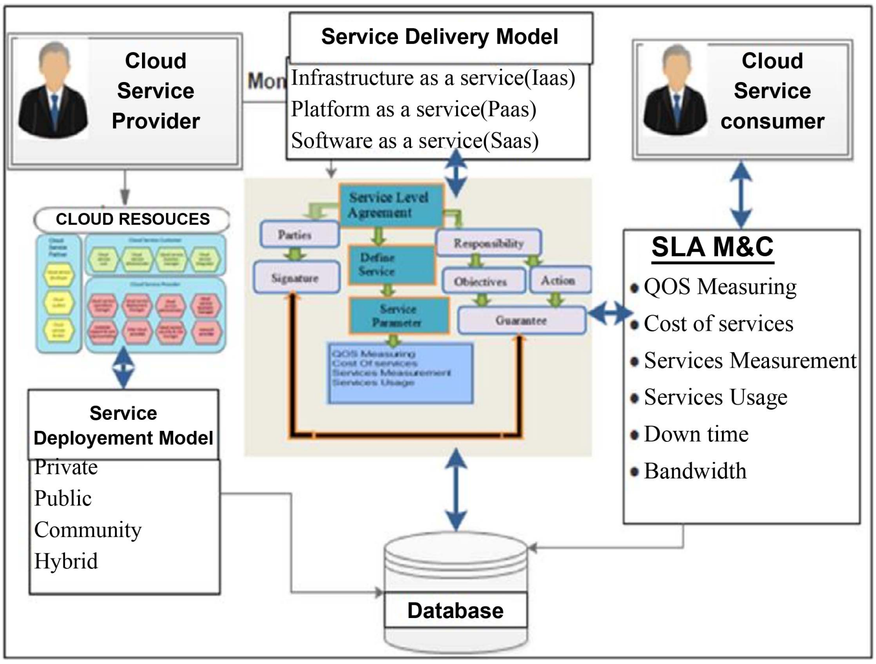

Figure 5. Proposed framework for M&C SLA for both server and client side.

because of different factors. So the service level agreement is violated. This framework was developed to solve this problem by adding a very important component which is known as SLA M&C (service level agreement monitoring and control) component. This component or the part of our framework used to monitor and control by comparing the services mentioned under the service level agreement and the services that are delivered to the cloud service consumer using different measurement as we demonstrated in an implementation of the prototype. The other important and our focus in this research is service level agreement component. It contains sub parts like service definition, service parameter, parties, signature, responsibility, actions, objective and guarantee. The cloud service consumer can control and monitor the service level agreement by reading the service level agreement document from the database/data center and compare with what services are delivering. This mechanism takes place by checking the quality of service features like cost of service, service measurement, service usage, down time and bandwidth.

Research Fining (Outcomes)

This research solved the gabs that are mentioned under the abstract and introduction part of this paper. We used three basic tools *i.e.* cloudSim for simulation purpose, MySQL server for creating data base for creating sample database to show the implementation and javaNetbeans for writing programing language to connect database with selected simulation (CloudSim). Our framework is increasing the quality of services by controlling and meandering the service level agreement. This mechanism is work for both cloud service providers and services consumer or server and consumer side. This prototype was implemented using four scenarios to evaluate its performance and the functionalities. 1) Checking the SLA under CSP whether the resources to the consumer in the same manner with promised. This means to check whether the delivered services are according to the service level agreement or not. So to do this we took the same values of the parameter fixed in database with the input data while checking. 2) To check if the request from cloud consumer is less than what expected in SLA. This scenario is the second one and it is used to check what if the requested from cloud consumer is less than what were expected in the service level agreement. 3) To check by increasing the request of cloud consumer more than what is expected in SLA. This is the third scenario and it was selected to check what if the request from cloud service consumer is the higher amount of services with greater parameter *i.e.* more virtual machine, host and ram which are greater than what were fixed in the service level agreement and 4) Using the SLA_M&C mechanism. This the last scenario was about the main important implementation of our research and what is needed by cloud service consumers and cloud service providers as well. Because both parties need benefit from the technology. Cloud service provider get payment from the cloud service consumer if the quality of the services are monitored and controller. Cloud service consumer also saved from buying resources instead they can access from cloud providers like infrastructure, platform and software as per use the services. This can be done if and only if the services and quality of the services are controlled and monitoring from both sides. They delivered to the consumer and the consumers also get benefit from the services. After execution it was observed that, in the first and the second cases there is no violation, but there is violation in the third and fourth cases. This implied that there are chances of the discrepancies in resources provided to make the selves beneficiary beyond SLA under CSP and CSC was discriminated while receiving services from CSP. SLA M&C was used to compare the service level agreement signed between the two parties cloud service provider and cloud service consumer and the features of quality of services. This component makes alarm to the cloud service consumers if the service level agreement is violated and the negotiation must be made with the cloud service provider to be continue as a business partner.

Figure 6 showed that the database contains the agreement between the service consumer and service provider before services provision. While we implemented the research framework, we used MYSQL database, CloudSim tool, Java as a programming language. So the above data base table showed different parameter that should be known during agreement like file size, number of virtual machine, number of host computer, and number of ram and so on. We used the above sample database table to show the role of Myqsl in this research. The column name indicated the parameter used in the research. Data Type indicated data types of the parameter indicated under column category.

This table indicates the sample of the service level agreement signed by the csc and csp before services are started to deliver. So both parties should follow this agreement during service delivery and payment of the services from the csc.

Table 1 showed that the parameter included the agreement between cloud service provider and cloud service consumer. We have taken length with 5000, File 400 size, output 200 size and CPU with 2 size. It is the input for the testing and measuring the results of our framework implementation.

Table 2 indicated that the results of all the four scenarios were put together using this table. Test case 1 indicated, the result of the first scenario mentioned under research findings or outcome. As a result if the consumer requests the services according to the service level agreement, the cloud service provider may violate the SLA. So our framework is designed to alarm to csc when this violation is happened. Under remark no violation means it should not be violated if csc requested according to SLA agreement. Test case 2 indicated that, if csc

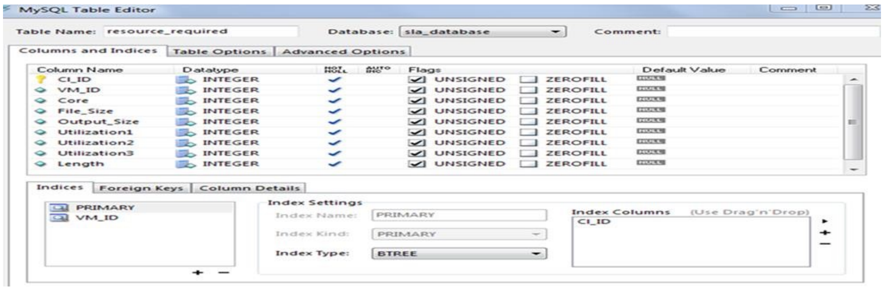

Table Name: resource_required
Database: sla_database
Comment:

Column Name	Datatype	NOT NULL	AUTO INC	Flags		Default Value	Comment
CI_ID	INTEGER	✓		☑ UNSIGNED	□ ZEROFILL	NULL	
VM_ID	INTEGER	✓		☑ UNSIGNED	□ ZEROFILL	NULL	
Core	INTEGER	✓		☑ UNSIGNED	□ ZEROFILL	NULL	
File_Size	INTEGER	✓		☑ UNSIGNED	□ ZEROFILL	NULL	
Output_Size	INTEGER	✓		☑ UNSIGNED	□ ZEROFILL	NULL	
Utilization1	INTEGER	✓		☑ UNSIGNED	□ ZEROFILL	NULL	
Utilization2	INTEGER	✓		☑ UNSIGNED	□ ZEROFILL	NULL	
Utilization3	INTEGER	✓		☑ UNSIGNED	□ ZEROFILL	NULL	
Length	INTEGER	✓		☑ UNSIGNED	□ ZEROFILL	NULL	

Indices: PRIMARY, VM_ID

Index Settings
Index Name: PRIMARY
Index Kind: PRIMARY
Index Type: BTREE

Index Columns (Use Drag'n'Drop): CI_ID

Figure 6. Shows SLA signed between CSC&CSP.

Table 1. Parameters that are fixed during agreement as a service level agreement.

CLOUDLETS	PARAMETER	SIZE
All Cloudlets	Length	5000
	File Size	400
	Output Size	200
	CPU	2

Table 2. The evaluation of all the scenarios.

Test cases	Cloudlets parameter	Request from CSC.	Allocated values By CSP.	Remark
	Checking the SLA under CSP whether the resources to the consumer in the same manner with promised services according to service level agreement			
1	Length	5000	5000	No violation
	File Size	400	400	No violation
	Output Size	200	200	No violation
	CPU	2	2	No violation
	To check if the request from cloud consumer is less than what expected in service level agreement			
2	Length	4000	4000	No Violation
	File Size	300	300	No Violation
	Output Size	100	100	No Violation
	CPU	1	1	No Violation
	To check by increasing the request of cloud consumer more than what is expected in SLA			
3	Length	6000	No service	Violation
	File Size	500	No service	Violation
	Output Size	300	No service	Violation
	CPU	3	No service	Violation
	The fourth scenario which is Used to desired the Mechanism SLA_M&C			
4	Length	6000	Our prototype Alarm to CSC	Violation
	File Size	400	Waiting until length will be corrected by CSC.	Violation
	Output Size	300	Our prototype Alarm to CSC.	Violation
	CPU	3	Waiting until CPU will be corrected by CSC	Violation

Note: csc: cloud service consumer and csp: cloud service provider.

requested fewer resources than mentioned in the SLA, the services should be provided by csp and the SLA should not be violated. In test case 3, if the csc requests more resources beyond the SLA, our framework ignores csc from the services and alarm notification to the csp. In test case 4, if some of the parameters are correctly requested and some are not correctly requested our designed framework alarmed to the csc to correct the request to obtain the services.

Sample code of implementation

```
Package SLA;
import java.text.DecimalFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.LinkedList;
import java.util.List;
import org.cloudbus.cloudsim.Cloudlet;
import org.cloudbus.cloudsim.CloudletSchedulerTimeShared;
import org.cloudbus.cloudsim.Datacenter;
import org.cloudbus.cloudsim.DatacenterBroker;
import org.cloudbus.cloudsim.UtilizationModel;
import org.cloudbus.cloudsim.UtilizationModelFull;
import org.cloudbus.cloudsim.Vm;
import org.cloudbus.cloudsim.VmAllocationPolicySimple;
import org.cloudbus.cloudsim.VmSchedulerTimeShared;
import org.cloudbus.cloudsim.core.CloudSim;
import org.cloudbus.cloudsim.provisioners.BwProvisionerSimple;
import org.cloudbus.cloudsim.provisioners.PeProvisionerSimple;
import org.cloudbus.cloudsim.provisioners.RamProvisionerSimple;
        int mips = 1000;
        long size = 10000; // image size (MB)
        int ram = 512; // vm memory (MB)
        long bw = 1000;
        int pesNumber = 1; // number of cpus
        String vmm = "Xen"; // VMM name
        // create VM
Vm vm = new Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm,
new
CloudletSchedulerTimeShared());
        // add the VM to the vmList
        vmlist.add(vm);
        // submit vm list to the broker
        broker.submitVmList(vmlist);
        // Cloudlet properties
        int id = 0;
        long length = 400000;
        long fileSize = 300;
        long outputSize = 300;
        UtilizationModel utilizationModel = new
        cloudletList.add(cloudlet);
  private static void printCloudletList(List<Cloudlet> list) {
    int size = list.size();
    Cloudlet cloudlet;
```

```
for (int i = 0; i < size; i++) {
        cloudlet = list.get(i);
        Log.print(indent + cloudlet.getCloudletId() + indent + indent);
        if (cloudlet.getCloudletStatus() == Cloudlet.SUCCESS) {
            Log.print("SUCCESS");
            Log.printLine(indent + indent + cloudlet.getResourceId()
                + indent + indent + indent + cloudlet.getVmId()
                + indent + indent
                + dft.format(cloudlet.getActualCPUTime()) + indent
                + indent + dft.format(cloudlet.getExecStartTime())
                + indent + indent
                + dft.format(cloudlet.getFinishTime()));
        }
    }
```

5. Conclusion

The worldwide computing mechanisms, model and technologies have been growing rapidly. The most dynamic and popular technology that has been transforming the Information and Communication Technologies (ICTs) and the work culture of the people is Cloud computing. The Cloud Service Providers (CSPs) have been hiding, or monopolizing the metering tools and ignoring the calculation and consideration of the service defects like downtime, service outage, poor performance and client service migration delays during overloads of the servers. The main discrepancies here are the violation of Service Level Agreement (SLA), and monopolization. After designing the framework using E-Draw Max, a prototype was designed and developed over CloudSim with java net beans. Also the MySQL was used for database handling. This framework was developed to resolve the aforementioned problem by inserting a newly evolved component SLA M&C (service level agreement monitoring and control). This component of the framework is used to monitor and control by comparing the services mentioned under the service level agreement and the services that are delivered to the cloud service consumer using different measurement as we demonstrated in an implementation of the prototype. Further the SLA that contains sub parts like service definition, service parameter, parties, signature, responsibility, actions, objective and guarantee parameters also included in the framework. Thus, the cloud service consumer can control and monitor the service level agreement by reading the service level agreement document from the database/data center and compare with what services are delivering along with the quality, availability promises, downtime, outage, etc. Thus this framework has developed a new knowledge base to empower the consumers with transparency in Monitoring and controlling while using Service level agreement.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Mell, P. and Grance, T. (2011) The NIST Definition of Cloud Computing. Recommendations of the National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-145

[2] Shekhawat, D.S. and Sharma, D.P. (2011) Cloud Computing Security through Cryptography for Banking Sector.

[3] Shekhawat, D.S. (2012) Hybrid Cloud computing in E-Governance: Related Security Risks and Solutions. *Research Journal of Information Technology*, **4**, 1-4.

[4] Dharmveer Yadav, D.S.B.K. (2017) Study of Intranet over Cloud. *International Journal of Innovation in Engineering and Technology*, **16**, 2.

[5] Sharma, D.P., Sharma, A.K. and Shekhawat, N.S. (2012) The Best Performance Practices in Project Management of SAP ERP Accomplishment. *International Journal of Information and Electronics Engineering*, **2**, 835-839.

[6] Sharma, D.P., *et al*. (2008) Convergence of Intranetware in Project Management for Effective Enterprise Management. *Journal of Global Information Technology* (*JGIT*)-*USA*, **4**, 65-85.

[7] Australian Prudential Regulation Authority (2018) Outsourcing Involving Cloud Computing Services.

[8] Patel, P. (2019) Service Level Agreement in Cloud Computing.

[9] Armbrust, M., *et al*. (2009) Above the Clouds: A Berkeley View of Cloud Computing. Technical Report, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley.

[10] Bahder, S. and Fathih, M. (2015) Cloud Computing for Distributed University Campus.

[11] Therese, L. (2015) An Effective Trust Based Access Control Services for Adaptive Monitoring Cloud Computing.

[12] Sharma, D.P. (2019) Client-Side SLA of Green Cloud Services Measuring and Monitoring.