

Role of Monitoring Service Level Agreement for Quality Enhancement over Cloud Computing Environment

Abel Adane Changare¹, Azath M.², Yitayish Lema³, Amin Tuni Gure⁴

¹Lecturer, Faculty of Computing & Software Engineering, AMIT, Arba Minch University, Ethiopia

²Assistant Professor, Faculty of Computing & Software Engineering, AMIT, Arba Minch University, Ethiopia

³Lecturer, Faculty of Computing & Software Engineering, AMIT, Arba Minch University, Ethiopia

⁴Assistant Professor, Faculty of Computing & Software Engineering, AMIT, Arba Minch University, Ethiopia

Email: ¹abeladi6@gmail.com, ²azathhussain@gmail.com, ³yitulema@gmail.com,

⁴atuni22@gmail.com

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 Cloud Service Consumer (CSC) was discriminated while receiving services from CSP.

Keywords: Cloud Computing, Service Consumer, Cloud Service Provider, SLA, SLA_CSC, SLA_CSP, SLA_M&Cetc.

1. Introduction

One of the main users concern refers to the validation and supervision of the quality of the provided services: users require greater levels of transparency through accurate and on-time delivery of SLA monitoring information. Nevertheless, monitoring is also fundamental for providers since SLAs are expected to be used by cloud vendors as their certification in order to establish themselves when

entering the competitive cloud market. Service Level Agreements play a central role in the service lifecycle, since by capturing service expectations and entities responsibilities they drive both engineering decisions at conception level (during for example service design) and operational decisions (during for example service usage and delivery).

SLAs enable participating entities to agree on what services will be offered, how will the services be delivered and who will be responsible for execution, completion, potential failures and privacy aspects (Mell, 2011). According to NIST the definition of cloud computing is Access, resource pooling, Rapid elasticity, Measured service and pay per use (Shekhawat, 2011), (Shekhawat D. , 2012), (Dharmveer Yadav, 2017), (Sharma, 2012), (Sharma D. e., 2016). 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.



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.

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 (Authority, 2018). Number of Researchers tried to do different researches on cloud service level agreement (Patel, 2019). According to (Mell, 2011) 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 Amazon Web 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 (Armbrust, 2009). 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 (Bahder, 2015) different problems were happened because of the violation of service level agreement.

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 (Bahder, 2015). Cloud Computing assign remote services with 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 (Therese, 2015). 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 (Sharma D. , 2019), (Therese L. , 2015), (Sharma A. A., 2008), (Serhat Bahder, 2015).

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

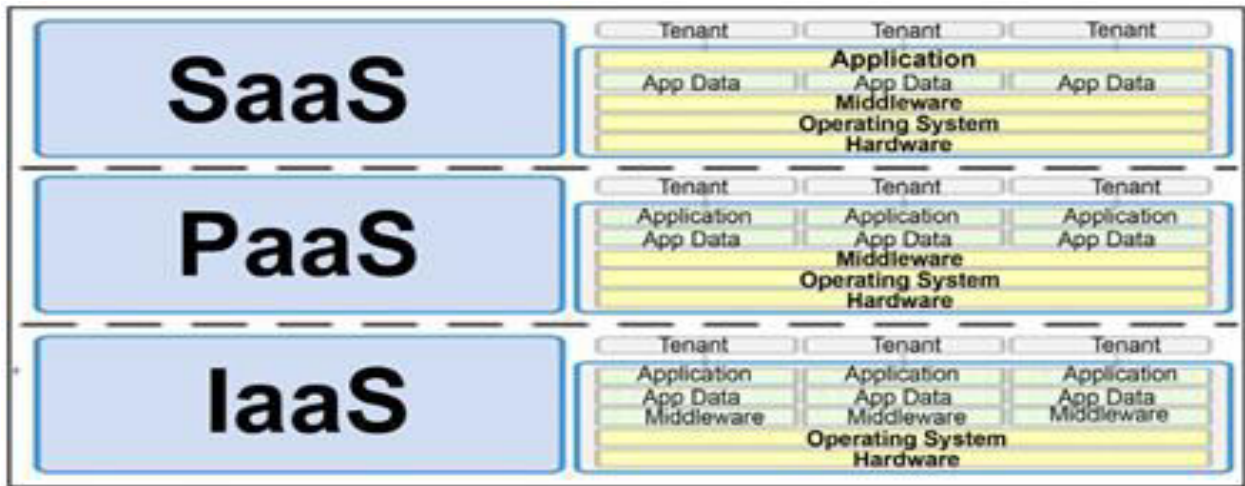


Figure 2: Cloud service delivery models

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, 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.

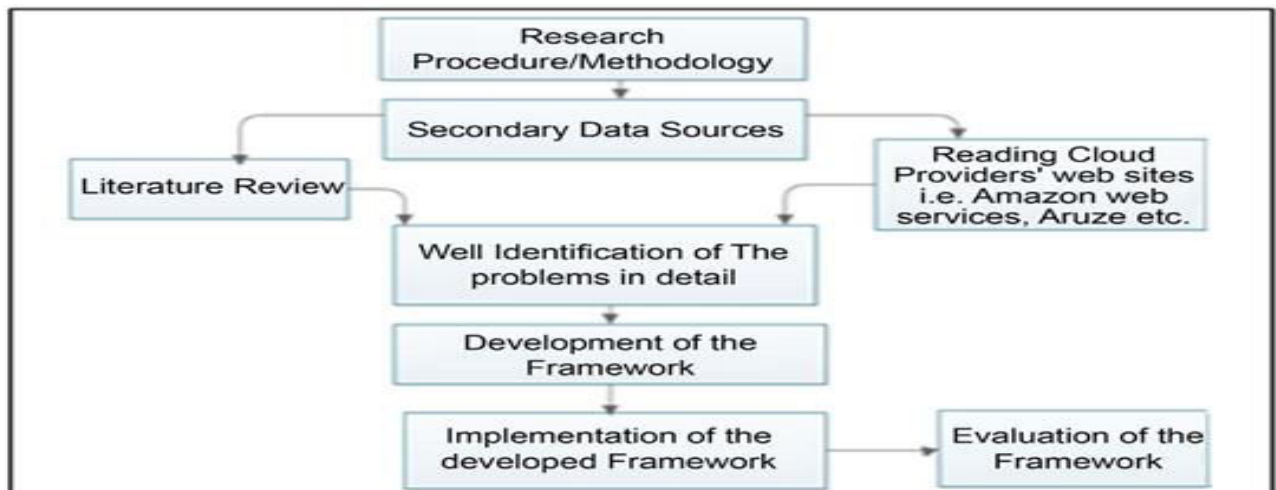


Figure 3: Research Methodology of the research

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 read 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.

3.1. Tool Selections

We had used different tools to complete this research. We used Microsoft office for different purposes like documentation purposes, E_Draw Max for developing framework and drawing different diagrams used in this research. We used Java Net beans for customizing cloudsims 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.

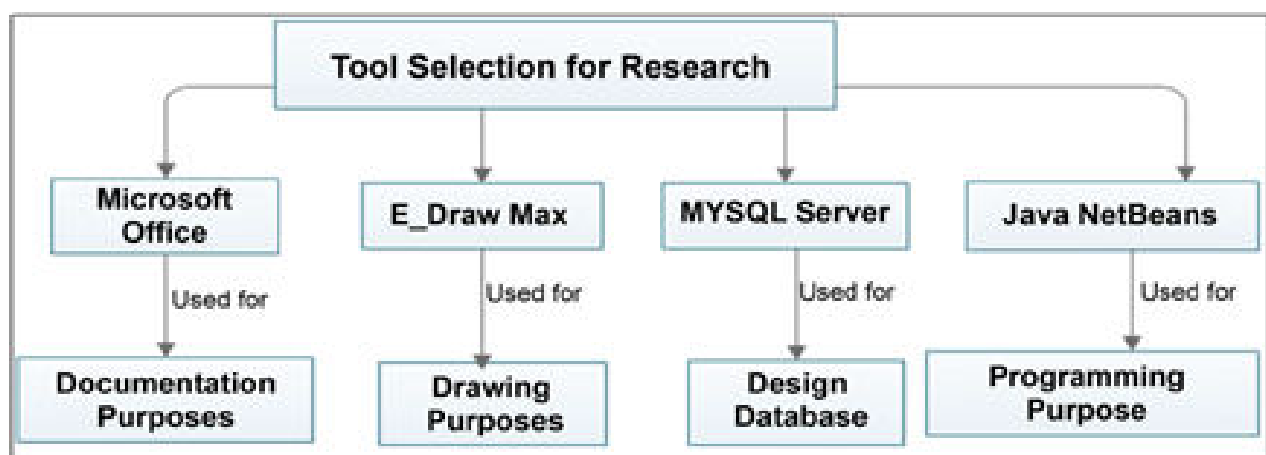


Figure 4: Tool selection for the research

4. Framework Design and Detail 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 shows the 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. This service level agreement is controlled and monitored by cloud providers and consumers not monitored and controlled only by cloud service provider.

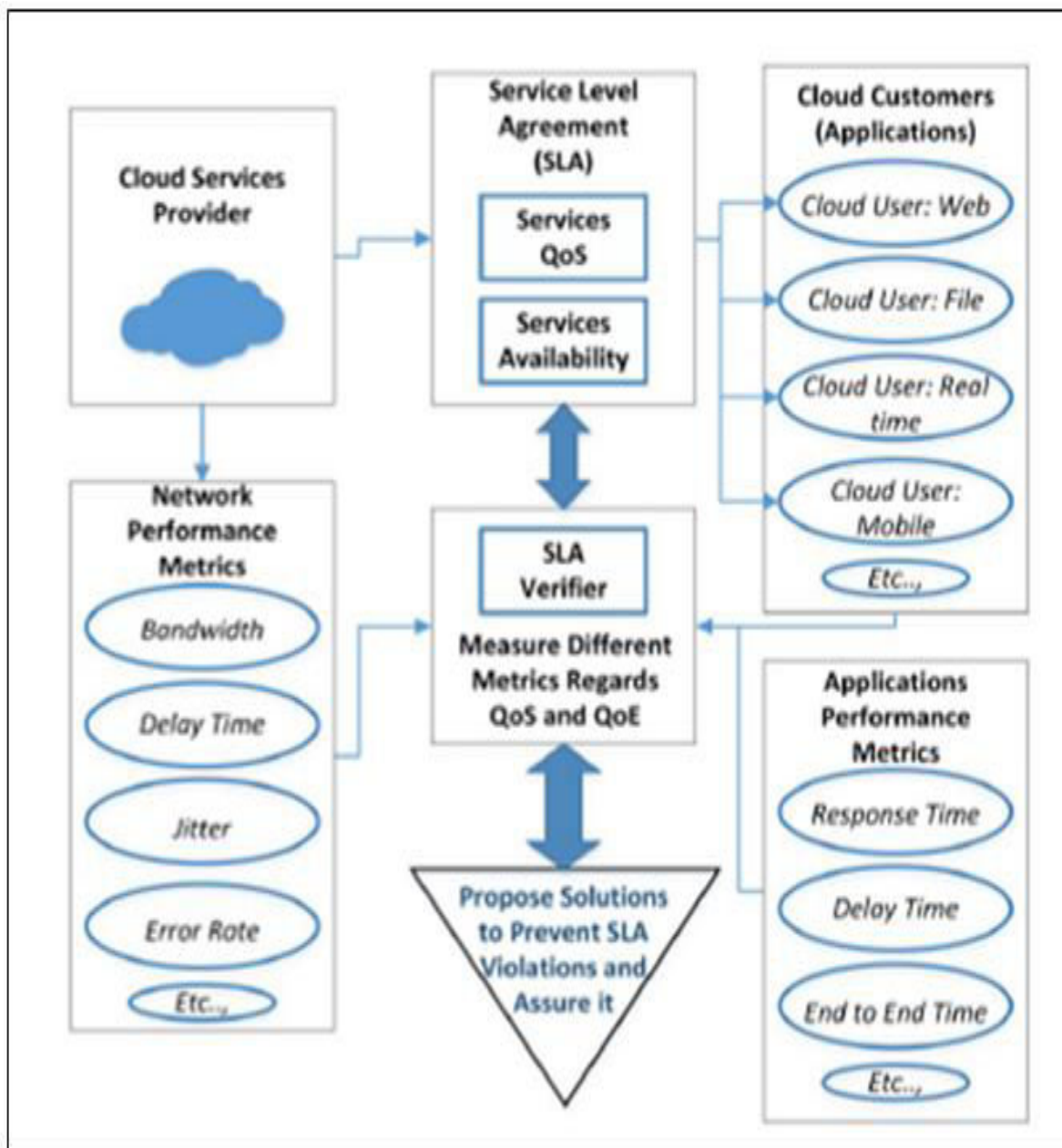


Figure 5: New Proposed framework developed for our research

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 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.

5. Research Findings (Result)

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 monitoring 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. 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 providers get payment from the cloud service consumer if the quality of services is 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.

Table 1: Shows how to create parameters using MYSQL server during agreement

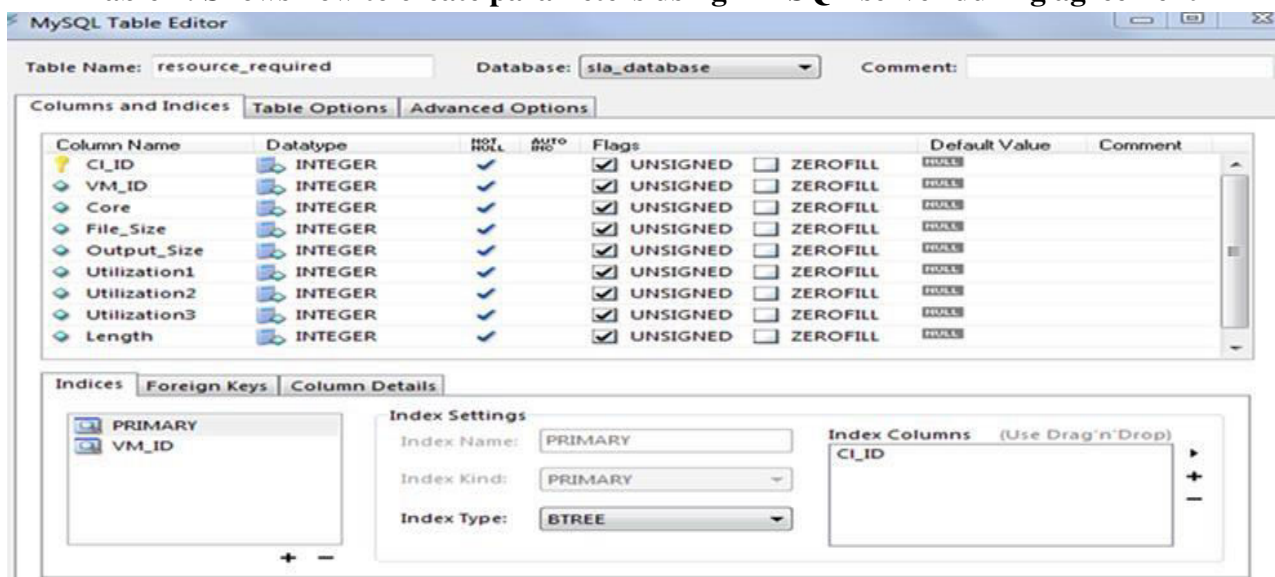


Table 1, showed the database contains the agreement between the service consumer and service provider before services provision. While we implemented the research framework, we used MYSQL database, CloudSim tool, Java as a programming language. So the above data base table showed different parameter 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 2: Parameters that are fixed during agreement as a Service Level Agreement

Cloudlets	Parameters	Size
All cloudlets	Length	6000
	File Size	500
	Output Size	300
	CPU	3

Table 2, shows the parameters included the agreement between cloud service providers and cloud service consumers. We have taken length with 6000, File 500 size, output 300 sizes and CPU with 3 sizes. It is the input for the testing and measuring the results of our framework implementation.

Table 3: The Evaluation Scenarios

Test cases	Cloudlets Parameters	Request from CSC	Allocated values By CSP	Remark
Checking when resources is requested according to SLA	Length	6000	6000	No violation
	File size	500	500	
	Output size	300	300	
	CPU	3	3	
To check if the request from cloud consumer is less than what expected in service level agreement	Length	3000	3000	No violation
	File size	200	200	
	Output size	100	100	
	CPU	1	1	
To check by increasing the request of cloud consumer more than what is expected in SLA	Length	7000	No service	Violation
	File size	600	No service	
	Output size	400	No service	
	CPU	4	No service	

Table 3, above indicated the result of our framework during evaluation. There are three cases in which we evaluated our developed framework. Table 3 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 1 and Test case 2 indicates cloud services consumers request according to the SLA and below Agreement as expectations. Test case 3 indicates if the consumer requested for service beyond the service level agreement it is violated. So no services at all in Test case3. Test case 4 is Control mechanism which was designed under the developed framework as shown using Figure 5: New Proposed framework developed for our research above.

Sample code

```
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.println(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()));
        }
    }
}
```

References

- Armbrust, et al. (2009). Above the Clouds: A Berkeley View of Cloud Computing, Technical Report No. UCB/EECS-2009-28. February 2009.
- Authority, A. P. (2018). Outsourcing Involving Cloud Computing Services. International Journal of Australia.
- Bahder, S. a. (2015). Cloud Computing for Distributed University Campus. International journal of Networking and communication .
- DharmveerYadav, D. (2017). Study of Intranet over Cloud. International Journal of Innovation in Engineering and, 16-26.
- Mell, P. a. (2011). The NIST Definition of Cloud Computing. Recommendations of the National Institute of. <https://doi.org/10.6028/NIST.SP.800-145>.
- Patel, P. (2019). Service Level Agreement in Cloud Computing. international Journal of computing Technology.
- SerhatBahder, M. F. (2015). Cloud Computing For Distributed University Campus.
- Sharma, A. A. (2008). Convergence of Intranetware in Project Management for Effective Enterprise Management, Journal of Global Information Technology (JGIT)-USA.
- Sharma, D. (2019). Client-Side SLA of Green Cloud Services Measuring and Monitoring. International Journal Of green cloud.
- Sharma, D. e. (2016). Convergence of Intranetware in Project Management for Effective Enterprise Management. Journal of Global Information Technology USA.
- Sharma, D. S. (2012). The Best Performance Practices in Project Management of SAP ERP. International Journal of Information and Electronics Engineering, 2, 835-839.

- Shekhawat, D. (2012). Related Security Risks and Solutions. Research Journal of. Hybrid Cloud computing in E-Governance:, 4-14.
- Shekhawat, D. a. (2011). Cloud Computing Security through Cryptography for Banking Sector. International Journal of cloud computing and technology.
- Therese, L. (2015). An effective Trust based access control services for adaptive monitoring cloud computing.
- Therese, L. (2015). An Effective Trust Based Access Control Services for Adaptive Monitoring Cloud Computing.