Scalable and Flexible SLA Management Approach for Cloud

SHAUKAT MEHMOOD*, AND ARIF IQBAL UMAR*

RECEIVED ON 26.08.2015 ACCEPTED ON 16.02.2016

ABSTRACT

Cloud Computing is a cutting edge technology in market now a days. In Cloud Computing environment the customer should pay bills to use computing resources. Resource allocation is a primary task in a cloud environment. Significance of resources allocation and availability increase many fold because income of the cloud depends on how efficiently it provides the rented services to the clients. SLA (Service Level Agreement) is signed between the cloud Services Provider and the Cloud Services Consumer to maintain stipulated QoS (Quality of Service). It is noted that SLAs are violated due to several reasons. These may include system malfunctions and change in workload conditions. Elastic and adaptive approaches are required to prevent SLA violations. We propose an application level monitoring novel scheme to prevent SLA violations. It is based on elastic and scalable characteristics. It is easy to deploy and use. It focuses on application level monitoring.

Key Words: Service Level Agreement, Service Level Agreement Analyzer, System Observer and Virtual Machine Manager, Cloud Platform, Elastic and Scalable,

1. INTRODUCTION

There is not a single agreed upon definition of the cloud computing. In literature several definitions are proposed by different researchers. Here we cite two definitions of the cloud computing.

First definition is, "A style of computing where massively scalable IT-enabled capabilities are de-livered as a service to external customers using Internet technologies" [1]. Second definition is, "A Cloud is a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers" [2].

In the Cloud environment, computing resources are rented to customers as services on payment according to as you go based method [3]. Therefore, computing resources in cloud must be available round the clock. The Cloud service provider offer services based on SLA. It stipulates the terms and conditions of services usage including the QoS parameters and details of penalties in case of SLA violations [3-4].

The cloud service providers must ensure reliability, scalability and availability of services to achieve on the demand computing resources. Competitive prices of the computing services is other major issue that influence the cloud business. Application layer monitoring is the key issue in cloud computing because one application

* Department of Information Technology, Hazara University Mansehra, Pakistan.

may run at multiple virtual machines or many applications may share the single VM (Virtual Machines).

The associated infrastructures of cloud computing provide significant scale and efficiency advantages. It enables users to replace their own IT infrastructures with large pools of computational and storage resources [5]. In the world of cloud computing, resource management is the core issue to handle workload fluctuations while maintaining QoS. Clients can claim using a pay-as-you-go ground on several combinations of computer cycles, network bandwidth, storage occupied and transaction executed [6]. The traditional resource usage and reservation-based charging policy cannot satisfy the application layer QoS because the SLAs are based on resource availability. Many cloud platforms such as Amazon's EC2 [7], are easy to use and support flexible cloud services. It is not easy to obtain necessary information for resource usages using the 3rd party cloud services, which put hindrance for conducting research on resource management for cloud computing. Establishing a reliable cloud platform and providing scalable resources with competitive prices can support cloud providers to meet customers' needs, increase revenue and profits, minimize penalties due to SLA violation. Ensuring SLA, especially at the application level, for different customers is not a trivial task. Developing an efficient application monitoring infrastructure to detect SLA violations of different customers and/or applications is a key issue for cloud providers. Moreover the environment should be easy to be customized to obtain dynamic resource information, e.g. usage of CPU (Central Processing Unit), memory, and link bandwidth for each VM. It must facilitate concept validation of diverse new research ideas and/or to conduct experiments for concrete data and empirical analysis.

Our proposed design and implementation is more elastic and easier to deploy. The platform is scalable, in which we can connect computers as a customer or as a provider at any time. We can assign the whole computer resources (CPU, Memory) or part of them to the environment as a cloud provider. Further, we can add new VMs from the SLA manager to a specific computer if it has extra resources.

2. MATERIALS AND METHOD

Zhang et. al. [8] present an adaptive approach based on novel policy to solve the problem. They presented a contract template embedded with a policy to handle the changes of service provision and participants' requirement. They illustrate how the policy enables adaptive SLA management. But the proposed approach lacks application level SLA management.

Landi et. al. [9] presented an architecture for the enforcement and validation of SLAs for mobile cloud services, discussed its positioning in the overall mobile cloud networking platform and its role across the different phases of the service lifecycle. They focused the mobile cloud services only.

Oliveira et. al. [10] proposed an architecture that consists of a novel approach for the monitoring and account of deviation of network SLA. They implement their architecture as open source engine. They concentrate on network SLA deviations only.

Emeakaroha et. al. [11] proposed a model called DeSVi. The proposed model monitors and detects violations of SLA infrastructure layers of cloud.

Sang and Eui [12] addressed the security issues, they provide a novel methodology of security risk assessment for security-service-level agreements in the cloud service based on a multi-dimensional approach depending on services type, probabilities of threats, and network environments to reach a security-SLA evaluation.

Motta et. al. [13] presented the elements of third party SLM, namely Cloud Service Registration Agent, Negotiation Agent, Compensation Agent, Comment Agent, Billing Agent and Service Monitoring Engine. They tested and analyzed third party SLM.

Antonescu et. al. [14] proposed a framework. Its main feature is to forecasting model for determination of best virtual machine that is allocated to host which will minimize the possible SLA violations, resource wasting and energy consumption. They focus forecasting the best virtual machine deployed on physical infrastructure that fulfills the need of consumer so that overcome the problem of SLA violations. But there is no mechanism to prevent violations at runtime.

Gadhavi et. al. [15] presented an architecture for engineering educational domain based on automated and reliable services deployment. This architecture enables consumer to select resources dynamically. In this model users are enabled to select services that are predefined and according to SLA. This model customizes the services according to requirements of user and deploys these services automatically. This model deploys resources efficiently but lacks any SLA violation detection method.

Mohamad et. al. [16] propose a framework for self-establishing an end-to-end service level agreement between a CSU (Cloud Service User) and CSPs (Cloud Service Providers) in a cloud networking environment by using brokerage service. They concentrated on QoS parameters for NaaS (Network as a Service) and IaaaS (Infrastructure as a Service) services. They proposed the model especially for videoconferencing that is challenging application in terms of bandwidth, delay and CPU resources usage, but lack of SLA violation detection.

Boniface et. al. [17] discuss the services dynamic provisioning using the GRIA SLA. The authors explained provisioning of services in cloud, grounded on agreed and signed Service Level Agreements to avoid violations. This methodology covers only Grid and not Cloud computing environment.

Koller and Schubert [18] explained proxy-like method for the management of quality of service and SLA. The proposed method covers the Web Service-Agreement. Thus, the agreed SLAs can be violated to explain some quality of service constraints that are necessary for a service to maintain while some specific customer interacts with it. This proposed method covers only Web services and does not support other types of applications.

Fruto and Kotsiopoulo [19] explain the methodology of the EU project “BREIN” [24] for the development of framework to elaborate features of computational Grids for the advance management of service level agreement in new areas of business domain to extend their use. But the “BREIN” covers the management of SLA in Grids environment only.

Comuzzi et. al. [20] explained the process for service level agreement establishment applied within the project named EU, SLA at SOI framework. While considering the two main requirements that are introduced by the Service Level Agreement establishment. The authors proposed the solution for SLA monitoring. The first requirement for SLA evaluation is availability of historical data, the second is capability valuation of monitoring term that is offered in SLA. This approach does ensure the main objective of SLA due to lacking of information to monitor the applications.

Dobson and Sanchez [21] discussed a unified ontology of QoS approach that is applicable to QoS based on Web-Based services selection, QoS adaptation and monitoring. But the authors do not explain how the deployment of application will occur and how different strategies regarding provision of resources will work.

Fu and Huang [22] present a model called GridEye a monitoring system that is service-oriented. This proposed model is flexible and furthermore this model contains a predicting algorithm. This algorithm predict the overall performance characteristics of resources. They explained how to monitor the deployed resources with proposed approach in Grid computing but they do not cover management of SLA.

Gunter et. al. [23] discussed NetLogger, a distributed system for monitoring that can collect and monitor network information. NetLogger's API is invoked by Applications to check the overload before and after some request/operation. It monitors network resources only.

2.1 Proposed Solution

We propose a scheme that is scalable and easy to deploy. We can connect computers as a customer or as a provider at any time. We can assign the whole computer resources (CPU, Memory) or part of them to the environment as a cloud provider. Further, we can add new VMs from the SLA manager to a specific computer if it has extra resources.

In the cloud computing environment the service procedure normally contains the following steps:

(1) Customers request for the services according to Signed SLA.

(2) Management framework validates the request.

(3) Virtual Machines configurator sets up Virtual Machines according to request and need.

(4) Application deployer allocates resources to the requested service.

(5) SLA Monitor framework detects violation.

Our scheme is composed of three main components namely: SLA Analyzer, SO (System Observer) and VMM (Virtual Machine Manager) as shown in Fig. 1.

The user requests the desired services by using the User Interface (Step-1). The SLA Analyzer receive that request and validates the user integrity and detail of request according to signed SLA (Step-2), if the status of the user and request are validated then the VMM creates the Cloud environment by deploying the pre-defined images of VMs on the Physical servers and make this environment accessible for provisioning of services (Step-3). After this the service request is forwarded to Application deployment (Step-4), for the execution of service it allocates the resources and deploy it in the Cloud setup.

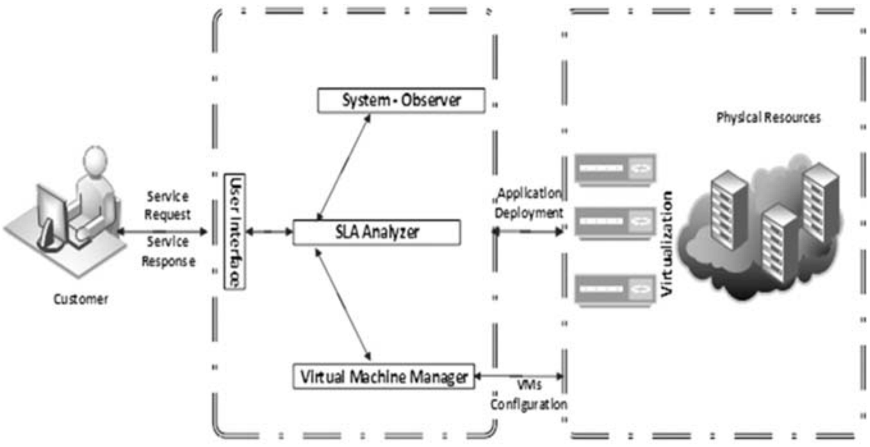


FIG. 1. PROPOSED SCHEME

When the service starts the execution in Cloud setup the SO monitor the execution of that service, collects monitoring matrices and passes these values to SLA Analyzer (Step-5). SLA Analyzer processes these values and detects SLA violations.

In addition to the above mentioned three main components here we discuss three subcomponents, these are; User Interface, VMs configuration and Application Deployment to present our scheme in detail. The SLA Analyzer module establishes the cloud computing environment, provision of cloud resources, and monitor SLA violations in the application layer.

The proposed architecture contains SO component for an elastic monitoring, that receives monitoring instruction from SLA analyzer component and sends the monitored data values back based on the SNMP (Simple Network Management Protocol) standard. To overcome the drawbacks of SNMP due to its complexity and overhead and platform related issues, we used Perl script to get domain information from Xentop.

The process of service provision and its management along with detection of SLA violation are supported in this architecture. The main component of SLA provision and detection is SLA Analyzer. First, SLAs between a customer and the provider are defined and stored in the SLA database (one of Data Dictionary). Various applications and threat and violation thresholds are defined in the SLA database. These values are used by the system observer detect violations of SLA and charge as per cost defined in SLA. In order to facilitate a large number of communication messages, the proposed system consists of a message queue of metric value pair using JMS and Apache ActiveMQ. MySQL database is used to store processed messages. Hibernate, Java based framework is used to store and retrieve information from database tables.

To set up the cloud environment we use different applications and tools. The technical detail of the set-up is as follows:

- Ubuntu **12.04.5**
- Windows 7 professional
- Xen 4.4
- NetBeans IDE 8.0.1
- Java Development Kit JDK 6.0
- MySQL Relational Database 5.6
- Apache Tomcat 8.0
- Jquery (Javascript framework)
- Xentop and Perl
- Putty and Dropbox

The first step is to set up the cloud that uses the Xen hypervisor 4.4, Ubuntu 12.04.5 and VMM. Xen hypervisor can run many instances on different operating systems, e.g. Linux, NetBSD and OpenSolaris, parallel on a single server. Due to its micro kernel design its memory footprint is small and has a limited user interface. It also supports Driver Isolation (the ability to permit the core device driver for a system to run inside of a VM), and Para virtualization. The Xen hypervisor is responsible for CPU scheduling, memory allocation, and interrupt handlings. In Xen, a running instance of a VM is called a domain here we named it admin-server for understanding and it manages VM creation, destruction, and configuration through the control stack [24]. VMM provides tools such as Viewer to offer lightweight interface for graphical display of live performance and resource utilization information. VMM also enables the new domains creation, configuration and adjustment of domain's resource.

After the installation of Xen hypervisor and utilities, we need to configure the Xend and compile Xen with library. The networking setup can be done by creating a bridge and installing and configuring dnsmasq. Then you will see the screen VM manager screen when running VMM in Fig. 2.

The Brain of the proposed scheme is SLA Analyzer. SLA Analyzer is a web based administrator portal to manage cloud environment. It provides functionalities like management of cloud service subscribers, VMs, SLAs, configuration values for applications, and records of customer's applications and SLA parameters. The SLA Analyzer deals with requested service and detects SLA violations. SLA Analyzer is central component and interacts with all others components. SLA violation management is achieved by receiving the monitored values from the System Observer agents that resides in the nodes where services execute. The design of SLA analyzer allows them to access the database where original signed SLA attributes are stored. With the pre-defined violation threshold and SLA database attributes the SLA Analyzer detects SLA violations. Threshold value indicates the performance least acceptable level of an application. These thresholds are created from the agreed and signed SLA document. With this information the system detects the SLA violations and calculates the penalties against these violations.

The final and a crucial component is cloud monitoring. Various options are available like CPU proc information, system log files, SNMP-Daemons and Xen-top. Reading and parsing CPU proc info and log files is not an easy job. Using SNMP Daemons is a good option if the daemons can be installed on each node to obtain data. Again, managing them is a challenging task, because every time on the front node one needs to install and configure it. A much easy way is to get the performance statistics from hypervisor only. That idea leads to Xen-top. It is a similar utility as the top command in Linux, supported by Xen hypervisor and provides lots of details about different VM. Additionally, it is accurate and details are directly provided from the hypervisor.

3. RESULTS AND DISCUSSION

For efficient monitoring we use Perl script. We defined Data Dictionary at different tables to store related information about VMs, users, and applications. For example, Applications Table contains information about applications running in the Cloud Platform. Xentop (nettx) provides aggregated bytes downloaded from each domain. Perl script logs readings after every 10 seconds (and configurable) in MySQL table. Hence, to find out the bandwidth, difference of current and previous nettx bytes is logged in MySQL. For each reading, current date

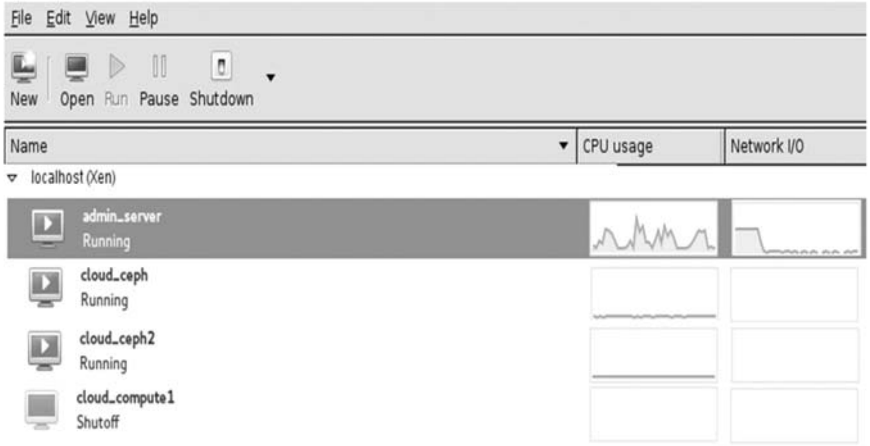


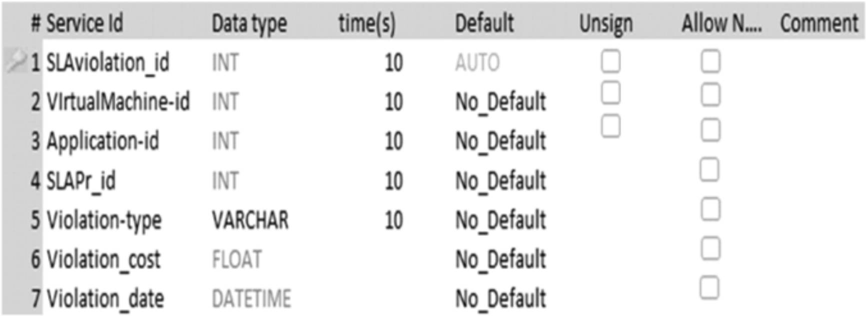
FIG. 2. VIRTUAL MACHINE MANAGER

and time are also logged in a tableget accurate readings. These readings are checked by same Perl script against defined service level agreement for bandwidth in SLA Manager for particular VM. Finally, SLA violations are logged in SLA Violations Table by the same script as live violation detection. The SLA Violations Table is composed of information about service level agreement violation, which is shown in Fig. 3.

Another Perl script checks violations of data (download) limit on each VM. Script iterates through each domain VM defined in SLA Manager and finds the sum of monthly bytes consumed by the cloud subscriber. These readings are checked against the pre-defined SLAs in SLA Analyzer for that particular VM and logged in SLA Violations table described above. Xentop utility was used to monitor the cloud environment. It provides information per domain. Perl was used for data extraction from Xentop, the basic idea was adapted from [25]. This script adds data after every 10 seconds into MySQL table created for each domain. The SLA violation detail table is shown in Fig. 4.

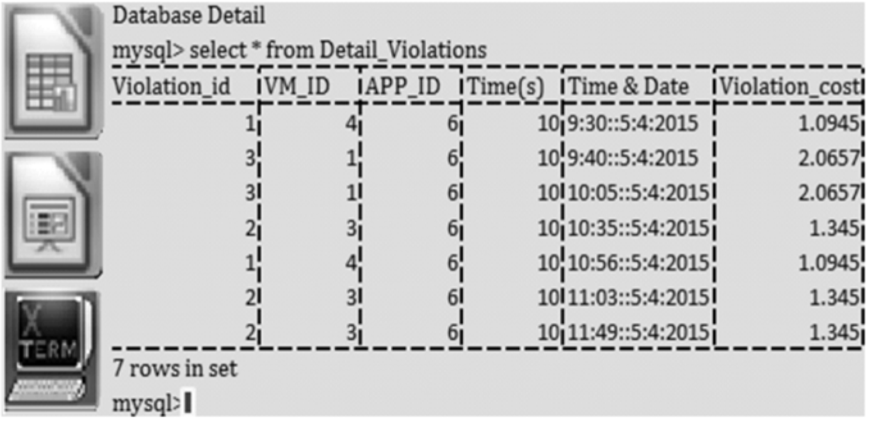
We can satisfy the customer to provide the log of violation as shown in Fig. 4. The cloud services provider pay the penalties against each violations. The cloud provider calculates the total cost of violations and can analyze the violation type that is in high rate. The total number of violation and their cost is shown in Fig. 5.

The ratio of violation may help the cloud service Provider to enhance his overall system performance. The detail of violations and cost of Fig. 5 in graphical form is shown in Fig. 6.



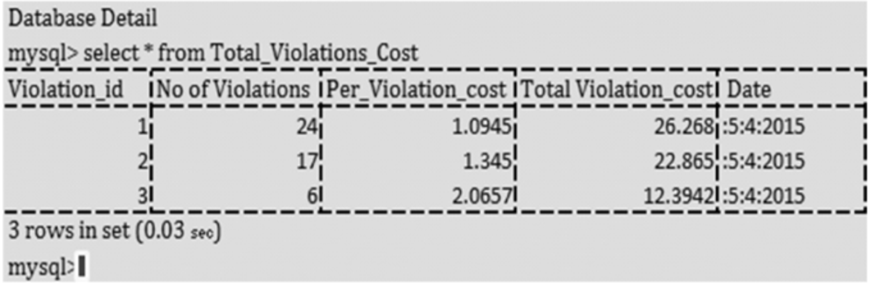
#	Service Id	Data type	time(s)	Default	Unsign	Allow N....	Comment
1	SLAviolation_id	INT	10	AUTO	□	□	
2	VIrtualMachine-id	INT	10	No_Default	□	□	
3	Application-id	INT	10	No_Default	□	□	
4	SLAPr_id	INT	10	No_Default		□	
5	Violation-type	VARCHAR	10	No_Default		□	
6	Violation_cost	FLOAT		No_Default		□	
7	Violation_date	DATETIME		No_Default		□	

FIG. 3. VIOLATION TABLE



```
Database Detail
mysql> select * from Detail_Violations
-------------------------------------------------------------------------------
Violation_id |VM_ID  |APP_ID |Time(s) |Time & Date       |Violation_cost|
-------------------------------------------------------------------------------
            1|      4|      6|      10|9:30::5:4:2015    |        1.0945|
            3|      1|      6|      10|9:40::5:4:2015    |        2.0657|
            3|      1|      6|      10|10:05::5:4:2015   |        2.0657|
            2|      3|      6|      10|10:35::5:4:2015   |         1.345|
            1|      4|      6|      10|10:56::5:4:2015   |        1.0945|
            2|      3|      6|      10|11:03::5:4:2015   |         1.345|
            2|      3|      6|      10|11:49::5:4:2015   |         1.345|
-------------------------------------------------------------------------------
7 rows in set
mysql>
```

FIG. 4. VIOLATION DETAIL



```
Database Detail
mysql> select * from Total_Violations_Cost
------------------------------------------------------------------------------------
Violation_id | No of Violations | Per_Violation_cost | Total Violation_cost | Date  |
------------------------------------------------------------------------------------
           1 |               24 |             1.0945 |               26.268 |:5:4:2015 |
           2 |               17 |              1.345 |               22.865 |:5:4:2015 |
           3 |                6 |             2.0657 |              12.3942 |:5:4:2015 |
------------------------------------------------------------------------------------
3 rows in set (0.03 sec)
mysql>
```

FIG. 5. SLA VIOLATION TYPE AND COST

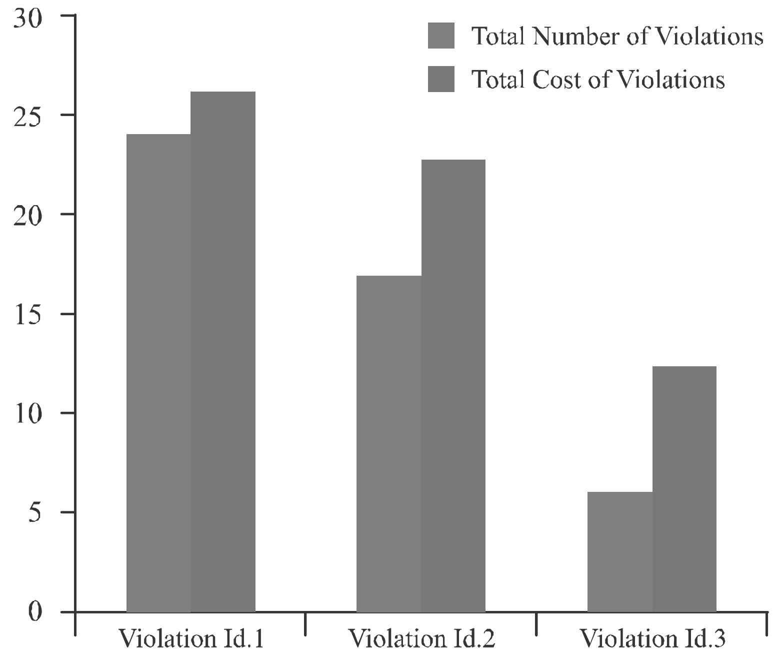


FIG. 6. VIOLATIONS AND COST

4. CONCUSION

This paper proposed a practical solution for establishing cloud computing environment and performing SLA violation detection. The proposed approach is easy to implement and could be configured dynamically. Our scheme is able to detect SLA violations at application level and provide the detail of violations along with their cost to the cloud service provider. The cloud service provider can provide the detail log of violations and penalties information to the customer for their satisfactions. Our designed scheme is valid for the single data centres. In future we will extend its working for multiple data centre.

ACKNOWLEDGEMENTS

The authors would like to thank Mr. Noor-ul-Amin, Department of Information Technology, Hazara University, Mansehra, Pakistan, for his valuable suggestions on the research which is published in this paper. Authors are also thankful to the Editorial Board, Mehran University Research Journal of Engineering & Technology, Jamshoro, Pakistan, for providing platform to publish our research. Authors also appreciate the wise guidelines given by the Referees/Experts.

REFERENCES

[1] Plummer, D.C., Bittman, T.J., Austin, T., Clearley, D., and Smith, D.M., "Cloud Computing: Defining and Describing and Emerging Phenomenon", Gartner, Inc. Retrieved September 25, 2008.

[2] Buyya, R., Yeo, C.S., and Venugopal, S., "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering it Services as Computing Utilities", 10th IEEE International Conference on High Performance Computing and Communications, pp. 5-13, 2008.

[3] Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., and Brandic, I., "Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility", Future Generation Computer Systems, Volume 25, No. 6, pp. 599-616, 2009.

[4] Comuzzi, M., Kotsokalis, C., Spanoudkis, G., and Yahyapour, R., "Establishing and Monitoring SLAs in Complex Service Based Systems", Proceedings of 7th International Conference on Web Services, 2009.

[5] Nathuji, R., et. al., "Q-Clouds: Managing Performance Interference Effects for QoS-Aware Clouds", EuroSys, 2010.

[6] Agarwala, S., et. al., "iCostale: Adaptive Cost Optimization for Storage Clouds", Proceedings of 4th IEEE Conference on CloudCom, 2011.

[7] Amazon, "Amazon Elastic Compute Cloud (Amazon EC2)", January, 2013, http://aws.amazon.com/ec2/.

[8] Zhenling, Z., Lejian, L., Hai, L., and Guoqiang L., "Policy-Based Adaptive Service Level Agreement Management for Cloud Services", Proceedings of 5th IEEE International Conference on Software Engineering and Service Science, pp. 496,499, 27-29 June, 2014.

[9] Landi, G., Metsch, T., Neves, P.M., Mueller, J., Edmonds, A., and Secondo, C.P., "SLA Management and Service Composition of Virtualized Applications in Mobile Networking Environments", IEEE Symposium on Network Operations and Management, pp. 1-8, 5-9 May, 2014.

[10] Oliveira, A.C., Chagas, H., Spohn, M., Gomes, R., and Duarte, B.J., "Efficient Network Service Level Agreement Monitoring for Cloud Computing Systems", IEEE Symposium on Computers and Communication, pp. 1-6, 23-26 June, 2014.

[11] Emeakaroha, V.C., Calheiros, R.N., Netto, M.A.S., Brandic, I., and De Rose, C.A.F., "DeSVi: Architecture for Detecting SLA Violations in Cloud Computing Infrastructures", Proceedings of 2nd International ICST Conference on Cloud Computing, 2010.

[12] Sang-Ho N., and Eui-Nam, H., "A Methodology of Assessing Security Risk of Cloud Computing in User Perspective for Security-Service-Level Agreements", 4th International Conference on Innovative Computing Technology, pp. 87-92, 13-15 August, 2014.

[13] Motta, G., Linlin, Y., Sfondrini, N., Sacco, D., and Tianyi, M., "Service Level Management (SLM) in Cloud Computing - Third Party SLM Framework", IEEE 23rd International Conference on WETICE, pp. 353-358, 23-25 June, 2014.

[14] Antonescu, A.-F., Robinson, P., and Braun, T., "Dynamic SLA Management with Forecasting using Multi-Objective Optimization", IFIP/IEEE International Symposium on Integrated Network Management, pp. 457,463, 27-31 May, 2013.

[15] Gadhavi, L., Korat, D., and Bhavsar, M., "Proposed Cloud Architecture for Automated and Reliable Service Provisioning of Engineering Educational Domain", International Conference on Engineering, Nirma University, pp. 1-7, 28-30 November, 2013.

[16] Hamze, M., Mbarek, N., and Togni, O., "Autonomic Brokerage Service for an End-to-End Cloud Networking Service Level Agreement", IEEE 3rd Symposium on Network Cloud Computing and Applications, pp. 54-61, 5-7 February, 2014.

[17] Boniface, M., and Phillips, S.C., Sanchez-Macian, A., and Surridge, M., "Dynamic Service Provisioning using GRIA SLAs", Proceedings of 5th International Workshops on Service-Oriented Computing, 2007.

[18] Koller, B., and Schubert, L., "Towards Autonomous SLA Management using a Proxylike Approach", Multiagent Grid Systems, Volume 3, No. 3, pp. 313-325, 2007.

[19] Frutos, H.M., and Kotsiopoulos, I., "BREIN: Business Objective Driven Reliable and Intelligent Grids for Real Business", International Journal of Interoperability in Business Information Systems, Volume 3, No. 1, pp. 39-42, 2009.

[20] Comuzzi, M., Kotsokalis, C., Spanoudkis, G., and Yahyapour, R., "Establishing and Monitoring SLAs in Complex Service Based Systems" Proceedings of 7th International Conference on Web Services, 2009.

[21] Dobson, G., and Sanchez-Macian, A., "Towards Unified QoS/SLA Ontologies", Proceedings of IEEE Workshops on Services Computing, 2006.

[22] Fu, W., and Huang, Q., "GridEye: A Service-Oriented Grid Monitoring System with Improved Forecasting Algorithm", Proceedings of 5th International Conference on Grid and Cooperative Computing Workshops, 2006.

[23] Gunter, D., Tierney, B., Crowley, B., Holding, M., and Lee, J., "Netlogger: A Toolkit for Distributed System Performance Analysis", Proceedings of 8th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 2000.

[24] Xen, "Management Tools &Interfaces", http://wiki.xen.org/wiki/Xen_Management_Tools#Management_tools _and_interfaces

[25] Maurer, M., et. al., "Adaptive Resource Configuration for Cloud Infrastructure Management", Future Generation Computer Systems, 2013.