ISSN: 2229-6956(ONLINE)
DOI: 10.21917/ijsc.2015.0137

AUTONOMOUS INTELLIGENT AGENT INDEMNIFICATION IN SLA (IAIS) ARCHITECTURE FOR EFFORTLESS MONITORING OF SLA VIOLATIONS

A. Kannaki[1] and J.M. Gnanasekar[2]

[1]*Department of Computer Science and Engineering, Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya, India*
E-mail: kannakianbu@ymail.com

[2]*Department of Computer Science and Engineering, Karpaga Vinayaga College of Engineering & Technology, India*
E-mail: jmg_sekar@yahoo.com

Abstract

Increasing demand for cloud computing is the reason for its ubiquitous growth in today's world. Enterprise consumers from large organization to small organization are preferring the services such as IaaS, SaaS and PaaS for their business growth with minimal cost because of on demand pay model. To employ such services for their organization, the cloud consumer and cloud service providers(CSP) has to consensus a agreement between them popularly known as Service Level Agreement(SLA). These SLA usually contains the type of service,Target Period, Quality of Service (QOS) properties, and Penalties called Service Level Objective (SLO) that has to be maintained during the service provision. This necessitates for the need to continuously monitor SLA violation to enforce penalties on the wronged one. Often this SLA management such as SLA monitoring and reporting is usually done by third party entity. But often these third party entities failed to ensure trust and compliance among the cloud consumers and CSPs. Henceforth in this paper, we propose a use of Intelligent Agent to monitor the SLO on behalf of the cloud consumers and report them of any discrepancies. We propose anIntelligent Agent Indemnification in SLA (IAIS) Architecture where the SLA management is entrusted to the dynamic, autonomous intelligent agent which is considered to be a substitute for human. We also measure some of the Key Performance Indicators and establish a threshold value to identify SLA violation based on threat threshold and will verify the same using simulation for effortless and efficient performance of the proposed framework.

Keywords:

Cloud Service Providers (CSP), Service Level Agreement (SLA), Quality of Service (QOS), Service Level Objective (SLO), Intelligent Agent Indemnification in SLA (IAIS)

1. INTRODUCTION

Cloud Computing has gained more importance because of its attractive features such as vast type of services ranging from development platform, software, hardware, to IT as a service etc., and on demand pay per need. The important entities involved in cloud computing are cloud consumer, cloud carrier, cloud broker, cloud service provider and cloud auditor. The main focused entities are cloud consumer and cloud service provider. To ensure service satisfaction and obligation, a formal contract is drawn between these entities known as Service Level Agreement. A typical SLA should contain the following components as in [1] namely:

- Objectives/Purpose
- Restrictions
- Target Validity Period
- Scope
- Entities involved
- Penalties
- Rewards

Among these components, penalties and rewards are must. A typical SLA should their service level in the form of attributes such as reliability, availability, rewards, penalties, performance, and so on. To guarantee such level of service, the entities involved should be able to measure and monitor the relevant service metrics.

Service Level Objectives are the key element in a Service Level Agreements (SLA), which usually contains the information on service quality and Key Performance Indicators. SLO can be measured using Key Performance Indicators (KPI), KPI can be quality of service, peak and average loads of work, the volume of demand at different times of delay and penalty for the cloud provider in case the provider fails to meet those requirements which is briefed in [2] and [3]. SLA monitoring is often based on computation and networks. Metrics such as response time, speed, throughput, utilization are computation based and jitter, bandwidth, capacity, delay are network based as explained in [4] and service metrics terms and formulas are described in [5].

SLA monitoring is often done by third party entities but they are done also by client based entities and providers based entities based on the requirement. Third party entities have often failed to meet the standards of trust and compliance in most cases. Consequently, we propose the use of dynamic and autonomic intelligent agents to measure SLA metrics and monitor SLA violation and enforce SLA penalties in this paper via Intelligent Agent Indemnification in SLA (IAIS) architecture. This architecture has the following five responsibilities.

- Discover apt Cloud Service Provider with required resources
- Evaluate those CSP's resources
- Negotiate the terms and conditions of SLA and establish mutual agreement
- Monitor to detect SLA violations and enforce penalties
- Assist to choose a new Cloud Service Providers

This paper is organized as follows. Section 2 describes the discussion on the related works and section 3 briefs about Intelligent Agent, section 4 describes the components of IAIS architecture and process involved, section 5 describes the experimental setup and results and section 6 provides the result comparison with other frameworks and finally section 7 describes the conclusion and future work.

2. RELATED WORKS

In this section, we will discuss in brief about the related works and their views on SLA violations in cloud computing.

According to Alhamad et.al [10], negotiation process in SLA framework plays main role and adversely have a cause-effect relationship with SLA monitoring. In this paper, they have also discussed about the possibilities of designing SLA using various technologies.

Jiacheng Yao [11] have deployed Web Service Level Agreement (WSLA) architecture for SLA monitor. Here the monitoring process is carried out by the third party. The author also reasons that semantics based web technology can be used to represent the perfect match for SLA violations.

Subsequently, Vincent [12] proposed a DeSVi architecture which implemented two level of violation detection from lower to higher levels. And also service availability (A) parameter is checked using Domain Specific languages (DSL). Vincent along with Brandic proposed a layered architecture where bottom up propagation of failures to the layer above is employed for SLA violation detection. Later on, Vincent again proposed a new architecture for automatic SLA violation detection named as CASViD, where SLA monitoring is done at the application layer as explained in [14].

Similarly, Salvatore developed and proposed a MOSAIC architecture in [15]. This paper is developed to provide better quality of service via cloud agency. Monitoring is handled by cloud agency. It also addresses the use of MAGDA which provides an agent based service for distributed computing.

Raj Kumar Buyya briefly explains about the management strategies for customer driven service management and computational risk management to sustain the service level agreement which is analysed in [16]. In addition, they have given an architecture approach for market oriented resource allocation and also global exchange of trades in clouds. Then Rajkumar Buyya along with Linlin Wu have paraphrased a survey on SLA provided by various services provider which is summarised in [17]. It also showcases the factors that are critical during the SLA Negotiation between the two parties. It then addresses the issues in violations of SLA and the penalties can be raised by consumer to the service providers.

3. INTELLIGENT AGENT

An intelligent agent (in simple terms known as Agents) is an autonomous entity which observes or gather information or responsible for providing some level of service without intervention by an external entity. They may vary from simple, goal based, model based, utility based to complex auto learning agents as in [6]. With respect to SLA management, we come across two important types of agents systems namely multi agents systems and mobile agents systems. These agents designated in [7] are responsible for their decision making, processes, interaction and behavior in general.

Multi agent systems are the system where multiple agents are deployed such that they together are responsible for work collaboration in this case SLA monitoring to achieve results. According to Foundation of Intelligent Physical Agents (FIPA) [18], there are two forms of communication is possible such as Agent Communication Language (ACL) and Knowledge Query and Manipulation Language (KQML). In ACL, communication is established via describing syntax through which interoperability between these agents are achieved. It is strongly based on semantics and logic. KQML is another important communication primitive which as created by DARPA knowledge sharing effort which was introduced with respect to ontologies in detail in [8]. It assumes that each intelligent agent has some knowledge assertions on another agent with which it communicates which is explained in [17].

Mobile agents as the name suggest moves from one host to another. These have gained popularity in recent years because of their mobility which allow them to move across to perform SLA management tasks. But there is a reason for concern regarding its security and administration which is a tedious one. IA is task oriented which has special characteristics like reactivity, autonomy, collaborative behavior, and knowledge based learning with decision making capabilities. Most important characteristics is the mobility, IA has the capability of navigation from one local host to another host. Intelligent Agents are introduced in this paper to check the indemnification of the SLO[3][6]. In [7] describes the advantages of agent based monitoring system. It address benefits like

- Increased server, services, and application availability
- Quick detection of server and operating system failures
- Quick detection of service and application failures
- Increased monitoring capabilities
- Increased administrative control

To support our proposal, we prefer multi agent system with ACL as communication primitive between these intelligent agents.

4. IAIS ARCHITECTURE

Intelligent Agent Indemnification in SLA (IAIS) Architecture is shown in Fig.1.

4.1 COMPONENTS OF IAIS

The underlying architecture of IAIS consist of three major entities namely Cloud Service Providers, Cloud Consumers and Intelligent Agent.

4.1.1 Cloud Consumers:

The Cloud Consumer interacts directly with our intelligent based on the requirements specified in the SLA. The cloud consumer makes a new service request via our intelligent agent by discussing the initial resource requirements. In some cases, the cloud consume may request a change in those requirements due to reasons such as resource forecast prediction, insufficient resources etc., Henceforth, these cloud consumers may be responsible for updating those changed requirements with the intelligent agent. To achieve this direct communication, we should define a user interface to receive those service requests. The Service request should be validated by the intelligent agent to ensure the right cloud consumers.

4.1.2 Cloud Service Providers:

The Cloud Service Providers abbreviated as CSP also should provide a compatible interface to communicate with our intelligent agent. These interface is responsible for two main process of intelligent agent namely SLA negotiation and migration. The Cloud service provider should contain information on cloud to generate the SLA templates. After discovery of suitable cloud service providers by the intelligent agent, the negotiation takes place between the intelligent agent and CSP.

4.1.3 Intelligent Agent:

As previously discussed, intelligent agents are an autonomous multi agent who is deployed on behalf of consumer with no loyalty to cloud service providers but act as an independent entity. These intelligent agents are initialized and deployed before all process begins as per cloud consumer interests. After initialization, an intelligent agent is responsible for the following five processes as per SLA lifecycle phases [8] namely Discovery, Evaluation, Negotiation, Monitoring and Migration.

4.2 PROCESSES OF IAIS

4.2.1 Discovery:

The Intelligent agent process begins with a service request from the cloud consumer after initialization of intelligent agent. Then the intelligent agent then searches all the cloud service providers and their SLA template and based on the requirements and suitability, selects the apt service provider. But in some cases, the cloud consume may request new service request. In such cases, the intelligent agent is responsible for rechecking with the same CSP and also request the SLA template for the changed request from other cloud service providers. Then based on best suitability, the same CSP or new CSP is chosen for the particular service request.

4.2.2 Evaluation:

After receiving the SLA templates from various cloud service provides, these templates are compared with each other based on the cloud consumer service request. For instance, if the cloud consumers request the use of software with minimum price and maximum validity time, then the SLA templates from various CSPs are analyzed and the best suitable candidate which provides minimum price and maximum validity time is selected. During evaluation, the elements necessary of SLA and suitable metrics are chosen by the intelligent agent.

4.2.3 Negotiation:

Once the suitable SLA template is chosen by the intelligent agent, then the intelligent agent begins the negotiation process with the corresponding cloud service providers. After negotiation the terms and conditions are carried over by the intelligent agent to the cloud consumer for approval. The cloud consumer if found reasonable, accept the terms of cloud service provider. If the terms are not acceptable, then the intelligent agent is informed and the next suitable cloud service provider is contacted for negotiation. This is because our system is mainly based on the welfare of cloud consumers rather than the cloud service providers. In some case, the cloud consumer may request the intelligent agent to renegotiate the terms and conditions for the services with the same cloud service provider. Hence three approval condition exists namely

- Approval
- Renegotiate and then approval
- Reject

4.2.4 Monitoring:

The most critical part of our IAIS process is SLA monitoring. The main job of our intelligent agent is to detect the SLA Violation and issue penalties in case of violation detection and reporting the same to the customer. SLA monitoring usually begin once mutual agreement is established.

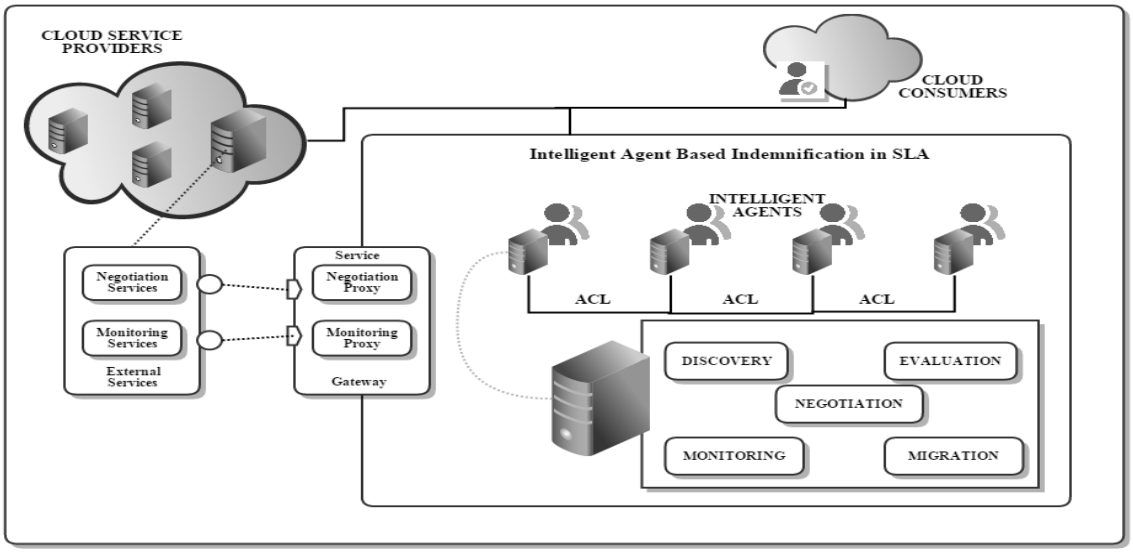

Fig.1. Intelligent Agent Indemnification in SLA (IAIS) Architecture

To establish SLA violation, we need to consider three criteria as discusses by Rana et.al in [9],

- "All or nothing" - It means all the SLO are met correctly
- "Partial"- It represents only the partial SLO are met
- "Weighted partial"- SLO is checked based on the threshold value

Our intelligent agent detects SLA unfulfillment based on the third criteria "Weighted partial" where the fulfillment of Service Level Objectives is checked based on the threshold value. Threshold value is nothing but a value used to indicate events that exceeded the predefined value. To understand the concept of threshold value, we will consider the following two service metrics namely Service Availability Time and Service Response Time. Service Availability Time is the time taken for the resource requested to be available. Service Response time is the amount of time taken by a service request to respond. Threat threshold are threshold value previously discussed and accepted value of predicted threat. For example, let's say tolerable availability time for a service request is predefined in the SLA with some value. This is the threat threshold for the service availability time for the service request. Suppose the availability time for the particular service exceeds the threat threshold value, then the SLA is violated.

A simple threshold algorithm to detect SLA violation is defined as follows.

Algorithm 1: Pseudo Code for threshold algorithm to detect SLA violation by intelligent agent

measuredValue ← Measured value of chosen service metric.
threatThreshold←The maximum allowed value of the chosen service metric.
SLAresult ← Result contains two options,namely "Violated" Or "Not Violated".
forvalueInstance ∈ measuredValue do
if valueInstance > threatThreshold
SLAresult:= Violated
Else
SLAresult:= Not Violated
return SLAresult

Based on the threshold value, the SLA violation is detected. As per SLA negotiation if any violation then penalty can be raised by the cloud consumer to the cloud service provider. The penalty over cloud service provider can be termination of the services, decrease in the payment, payment decline further and etc. Once the SLO violation is identified by the IA it will be saved in log information, and the information is sent to the cloud consumer, then the cloud consumer will raise the penalty to the respective cloud service providers.

4.2.5 Migration:

If the Cloud Consumer decides to change the cloud service provider due to SLA violation, then the intelligent agent is informed of the same. Then the intelligent agent proceeds with migration of service from old CSP to a newly chosen CSP by beginning the process from the beginning.

The communication between the intelligent agents is based on the standard of ACL whereas the communication between the cloud consumer and the intelligent agent and the cloud service provider and the intelligent agent is based on standard network management protocols such as Simple Network Management Protocol (SNMP), Remote Network Monitoring (RMON) based on the cloud consumer service request. The cloud service provider and intelligent agent can also request services such as negotiation and monitoring services externally where the cloud service provider accesses the external services directly but the intelligent agents access the external services via network gateway.

5. IMPLEMENTATION

The Proposed architecture is implemented using a simulator to define our scope, purpose and results. The experimental setup explains about the simulation environment and a sample scenario where SLA is violated and the next part explains in brief about the results achieved form our proposed architecture and other frameworks to analyze the performance of our architecture.

5.1 EXPERIMENTAL SETUP

The Simulation environment consist of three modules namely Cloud consumer module which is responsible for simulation the request for cloud resources. The second module is the intelligent agent module which is responsible for receiving service request, discovering cloud service providers, evaluating CSP, establishing SLA and negotiating terms and conditions and agreement is reached and the service is provided by the CSP to the cloud consumer. The intelligent is also responsible for monitoring the SLA violation based on our threshold algorithm and penalties are enforced if SLA violation occurs. The intelligent agent is also responsible for migrating to new CSP. The intelligent agent is implemented using JAVA and communication protocols such as SNMP, ACL are implemented using Java library in our simulation environment. Then cloud service provider module is simulated and resource instances such as virtual machine instance are generated. The sample scenario is given below where the service request is generated by the Cloud Consumer and forwarded to the intelligent agent is shown in Fig.2.

Service Request
Num. of Servers = (3)
CPU GHz = (1.5-3.0)
Storage (GB) = (2000-*)
Traffic(GB) =(1-*)
Operating Sys = <Windows, Linux>
Availability Time = [.5-.6 ms)
Price (EUR) = [0 - 1000)
Response Time = [.5 - .6 ms)

Fig.2. Sample Cloud Consumer Service Request

Then the intelligent agent module request SLA templates for the service request from different CSPs. Then the intelligent agent module proceeds with setting up an agreement. After SLA is established, then the intelligent agent monitors the services provided

by the CSP to the cloud consumer. We will consider the availability time and response time as the Service Level Objectives.

$$Availability\ Time = \frac{Uptime}{Uptime + Downtime} \quad (1)$$

The Eq.(1) is used to calculate the availability of a system where uptime is the time when the system or service is available and downtime is the time when the service is not available.

$$Response\ Time = \frac{Total\ Time\ taken\ by\ the\ service\ request\ to\ respond}{No\ of\ request\ raised} \quad (2)$$

The Eq.(2) is used to calculate the response time for a service request.

The above equations are used to measure the availability and response time of a service request which is then compared with the threat threshold values of the service request and SLA threshold algorithm is called by the intelligent agent to check the SLA violation and the results for the sample service request of Fig.2 is shown in the table below.

Table.1. SLA violation detection

SI. No	SLO	Measured Value(in milliseconds)	Threat Threshold Value(in milliseconds)	SLA result
1	Response time	0.42	0.5-0.6	Not Violated
2	Availability time	0.59	0.5-0.6	Not Violated

5.2 COMPARISON RESULTS BETWEEN AGENT SYSTEM AND OTHER FRAMEWORKS

Previously Third party system is used to perform auditing for the Consumer based on the requirement provided by him. But it's not secure in trusting a third party who evaluates the client's personal data. Client also cannot monitor the system at regular system, so agent based system is deployed on behalf of the client which is self-configurable and has capability of decision making in complex system. Hence this paper has initiated agent based SLA violation check on behalf of client and store in log file. IA based SLO violation check is much better compared to third-party system because agents are deployed by the consumer itself during the SLA negotiation process. IA are created based on credentials hence IA are found to be more secure compare to the third party auditing systems.

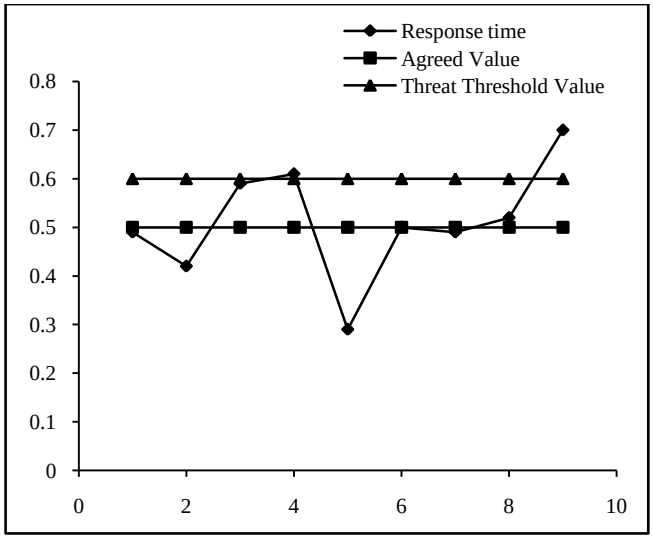

Fig.3. Analysis of our IAIS SLA indemnification for response time service metric

To analyze our IAIS architecture results, we have simulated a series of cloud consumer service requests and used it to check the effectiveness of SLA violation detection. In the below figure, the x axis means the service request number and the y axis means the response time in milliseconds. The agreed value of response time is 0.5 milliseconds. The threat threshold value for response time is 0.6milliseconds. Then the service response time for all the agreed service request is monitored and logged by the intelligent agent. Then they are checked against the threat threshold value to detect SLA violations. The 2 CSPs service request violation has been detected and the results are informed to the cloud consumer and SLA violation penalties are issued for those CSPs.

6. CONCLUSION

We have proposed an autonomous Intelligent Agent based SLA management and Indemnification in case of SLA violation in cloud computing. This paper also address the major benefits of deploying intelligent agents based system rather than third party system which resulted in increased cloud consumer welfare and no loyalty for the cloud service provider. In our proposed architecture, we also introduced a simple threshold algorithm for SLA violation detection and implemented the same via intelligent agent to enforce penalties on the cloud service providers and reduced the effort with easy migration to new service cloud provider. Dynamically updating and deleting of SLO in SLA resulting in dynamic SLA management using IA will be considered for future work with tamper protection against the dynamic SLA which was a major issue as discussed in [19]. Therefore tamper protection of dynamic SLA will be achieved using some suitable data protection scheme as an additional extension to our future work.

REFERENCES

[1] K. Stamou, V. Kantere and J. H. Morin, "SLA data management criteria", *Proceedings of IEEE International Conference on Big Data*, pp. 34-42, 2013.

[2] F. John Reh, "Key Performance Indicators (KPI)", White Paper, 2013, http://Management. About.Com/Cs/Generalmanagement/A/Keyperfindic.Htm.

[3] Yi Wei and M. Brian Blake, "Service-Oriented Computing and Cloud Computing Challenges and Opportunities", *IEEE Internet Computing*, Vol. 14, No. 6, pp. 72-75, 2014.

[4] Vinod Venkataraman, Ankit Shah and Yin Zhang, "Network-Based Measurements on Cloud Computing Services", http://www.cs.utexas.edu /~vinodv/ files/cc-measure.pdf

[5] A. M. Aljohani, D. R.W. Holton, I. U. Awan and J. S. Alanazi, "Performance evaluation of local and cloud deployment of web clusters", *14th International Conference on Network-Based Information Systems*, pp. 274-278, 2011.

[6] Jeffrey M. Bradshaw, "*Software Agents*", MIT Press, pp. 3-46, 2007.

[7] H. Kaindl, M. Vallee and E. Amautovic, "Self-Representation for Self-Configuration and Monitoring in Agent-Based Flexible Automation Systems", *IEEE Transactions on Systems, Man and Cybernetics: Systems*, Vol. 43, No. 1, pp. 164-175, 2013.

[8] Chen-Yu Lee, K. M. Kavi, R. A. Paul and M. Gomathisankaran, "Ontologies of Secure Service level agreements", *Proceedings of IEEE 16th International Symposium on High Assurance Systems Engineering*, pp. 166-172, 2015.

[9] O. F. Rana, M. Warnier, T. B. Quillinan, F. Brazier and D. Cojocarasu", Managing Violations in Service level agreements", *Proceedings of the 5th International Workshop on Grid Economics and Business Models*, pp. 349-358, 2008.

[10] M. Alhamad, T. Dillon and E. Chang, , "Conceptual SLA framework for cloud computing", *Proceedings of IEEE International Conference on Digital Ecosystems and Technologies*, pp. 606-610, 2010.

[11] Jiacheng Yao, "On-Demand Optimal Cloud Service Provisioning Composition across Multi-cloud", *Proceedings of Fifth International Conference on Computational and Information Sciences*, pp. 1566-1569, 2013.

[12] Vincent C. Emeakaroha, R. N. Calheiros, M. A. Brandic and C. A. De Rose, "DeSVi: architecture for detecting SLA violations in cloud computing infrastructures", *Proceedings of the 2nd International ICST Conference on Cloud Computing*, 2010.

[13] I. Brandic, V. C. Emeakaroha, M. Maurer, S. Dustdar, S. Acs, A. Kertesz and G. Kecskemeti, "LAYSI: A Layered Approach for SLA-Violation Propagation in Self-Manageable Cloud Infrastructures", *Proceedings of IEEE 34th Annual International Conference on Computer Software and Applications Conference Workshops*, pp. 365-370, 2010.

[14] Vincent C Emeakaroha, T. C. Ferreto, M. A. S. Netto, I. Brandic and C. A. F. De Rose, "CASViD: Application Level Monitoring for SLA Violation Detection in Clouds", *Proceedings of IEEE 36th Annual on Computer Software and Applications Conference*, pp. 499-508, 2012.

[15] Salvatore Venticinque, Rocco Aversa, Beniamino Di Martino, Massimilano Rak and Dana Petcu, "A cloud agency for SLA negotiation and management", *Proceedings of Euro-Par Parallel Processing Workshops, Lecture Notes in Computer Science*, Vol. 6586, pp. 587-594, 2011.

[16] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg and Ivona Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility", *Future Generation Computer Systems*, Vol. 25, No. 6, pp. 599-616, 2009.

[17] Linlin Wu and Rajkumar Buyya, "Service Level Agreement (SLA) in utility computing systems", *Performance and Dependability in Service Computing: Concepts, Techniques and Research Directions*, pp. 1-25, 2011.

[18] Bobroff, Norman, Andrzej Kochut and Kirk Beaty, "Dynamic placement of virtual machines for managing SLA violations", *Proceedings of IEEE International Symposium on Integrated Network Management*, pp. 119-128, 2007.

[19] M. R. Meybodi, "Decreasing Impact of SLA Violations : A Proactive Resource Allocation Approach for Cloud Computing Environments", *IEEE Transactions on Cloud Computing*, Vol. 2, No. 2, pp 156-167, 2014.