Generalized SLA Enforcement Framework using Feedback Control System

Mussadiq Abdul Rahim, Irfan Ul Haq, Hanif Durad
Department of Computer and Information Sciences
Pakistan Institute of Engineering and Applied Sciences
Islamabad, Pakistan
mussadiq.ar@gmail.com, {irfanulhaq, hanif}@pieas.edu.pk

Erich Schikuta
Research Group Workflow Systems and Technology
University of Vienna
Vienna, Austria
erich.schikuta@univie.ac.at

Abstract— **Cloud computing has emerged as powerful technology with various use cases in different environments. The goal of these use cases is to provide services to users on demand. These use cases vary from application to application, but service level agreement (SLA) management plays an important role in all of them. SLA management is an essential part of service provisioning environment. Automated SLA management is a necessary part of large production environments, where business heavily depends on customer satisfaction. In this paper we discuss a generalized SLA management framework and propose a generic framework for automated SLA enforcement based on feedback control system. We have selected various SLA metrics, describe inputs and outputs for those metrics and we propose enforcement methodology based on control theory. We define formal model based on control theory for availability metric. It is seen that feedback control automates the SLA enforcement with accuracy. Multiple SLA metrics can be enforced using this approach.**

Keywords—cloud computing, SLA, control, feedback control

I. Introduction

Cloud computing environment gives us a service provisioning environment. Each service is well defined between user and provider in form of a SLA. SLAs form a contract between service user and service provider, for provision of a service. There are various standards which define how SLAs may be defined formed e.g. WS-Agreement [1]. It defines the required type, attributes and quality of service (QoS) which are demanded by service user from service provider against the required service. Each functional requirement is defined as a guarantee term and a service level objective (SLO) is associated to that term. Services are measured using SLA metrics, these metrics define different quantitative and qualitative aspects or attributes and how they are measured. Service objects are monitored and measured and those measured values define current state or evaluated condition of system. SLA mapping defines how a term is mapped on service object and its functions. In case of a deviation from defined terms a SLA violation is triggered. On occurrence of a SLA violation, a decision is made how to adjust the underlying resources to meet the terms and if or may the service provider be penalized or service user be compensated for the incident. SLA Management also includes negotiation, agreement, and entire cycle till the expiration of SLA or decommissioning of service.

SLA enforcement is a major part of SLA management framework. In large cloud setups automated SLA enforcement is a requirement to ensure customer satisfaction to maximize profit. SLA enforcement is a continuous process it prevents SLA violation by continuously monitoring and adjusting service objects or resources in such manner SLA terms are fulfilled.

In this paper, section II, we discuss related work focusing on different aspects of SLA management framework. Different aspects involve SLA formation, SLA applications in various kinds of applications, privacy, SLA-Views, SLA-Choreographies and validation approaches. We divide section 0 into two parts, in first part we discuss a generalized SLA framework model, and in later part we discuss control theory specifically feedback control systems. In section IV, we map different SLA metrics with input and outputs of management system. Later on in this section we propose a formal model for SLA management targeted to SLA enforcement using feedback control system. This new approach enforces SLA metrics using feedback control system. In last sections V and VI we show results of simulation for availability metric and conclude the paper with a room for research in future.

II. Related Work

We study different kinds of frameworks which cover whole or some part of SLA management. [1][3] define new languages of representing SLAs, RBSLA [1] is targeted to describe the SLA in logic and rule form, whereas SCOL [3] utilizes XML and XPath with aim of mapping data and configuration in efficient manner. Some frameworks are specific to applications as [4][5]. Reference [4] is designed for database as a service (DaaS) where it separates the SLA for application and SLA for infrastructure. They developed a benchmark for testing DaaS on throughput and replication delay parameters. Also [5] is application specific to network services, they analyze the Ponder language when used for network SLA management. Framework [6] divides SLAs into a hierarchy for definition of multi-level SLA management. Business, Software, Infrastructure SLAs work at different levels. They define a technical architecture and assign responsibility to components which are used. Frameworks [7][8] are part of Foundations of Self-Governing ICT Infrastructures (FoSII) and both suggest method for monitoring for SLA violation. Framework [7] is based on three components with separate functions and [8] defines five layers where it interacts with SLA manager and through these five layers SLA from a user is automatically processed and monitored. Solution [9] defines a formalized privacy model for SLA-Views and formal description of hierarchical SLA-Choreographies based on SLA-Views in Business Value Networks. Solution [10] defines loosely coupled model for SLAs between cloud components. It defines a new validation approach which functions on higher level goals e.g. business rules. Solutions [11][12] defines aggregation model for SLAs which enables automation of hierarchical aggregation of SLAs. Solution [13] defines a mapping between monitoring and enforcement components to make SLA management unsupervised in a manner. Framework [14] defines SLA mapping, as SLAs are defined over SLA templates, but templates vary on user and provider sides, this mapping bridges gap between differently described SLAs. Solution [15] defines an analytical model forecasting the probability of finding matching providers for web service negotiations based on quality of service parameters. There is a room for solution for generalized framework which fits all shapes and sizes.

III. SLA MANAGEMENT AND CONTROL THEORY

A. *Generalized SLA Framework*

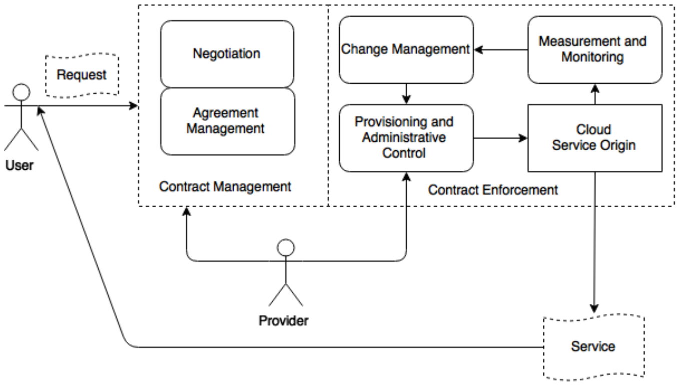

Fig. 1. Generic SLA Framework

A framework may define some or all components from generalization as provided in Fig. 1. We divide the generalized SLA Management framework into two major parts, first is contract management which handles SLA request which is based over some SLA template and is further negotiated and this part manages the agreement and catalogs it. Second part namely contract enforcement, handles the SLA lifecycle. SLA lifecycle is divided into four parts, first part, cloud which is generally the service origin it consists of the service objects. Second part, measurement and monitoring is a separate system or component of cloud which provides the measured values against each SLA metric. Third part, change management is the core part of a framework which continuously monitors the measurements provided by prior component and SLA terms for SLA violations. It may generate an alarm against a SLA violation and perform changes in quantity or quality of service being provided to prevent violation from occurring. Fourth component, provisioning and administrative control allows the service provider to induce changes as calculated by prior component. Whereas in automated SLA enforcement itself applies changes to the service.

B. *Feedback Control System*

Control theory makes it possible to model a physical, mechanical, digital system and other. The model based on the mathematics of the system utilizes varying states of system and target values to achieve target in a manner. Feedback control is central to managing computing systems and networks. For example, feedback (or closed loop systems) is employed to achieve response time objectives by taking resource actions such as adjusting scheduling priorities, memory allocations, and network bandwidth allocations [16]. There are three kinds of control objectives. Regulatory control, ensures the output of system is near to the input (reference value) of system. Disturbance rejection, makes sure that the noise, distortion or any kind of disturbance affects the output of system, least. Optimization, choose the best value or solution independent of the reference value [17].

A control system is considered stable, if there is a bounded output against a bounded input. It is accurate if its output converges to the input (reference value). Systems settling time is whether short or not, better system will have shorter settling time, settling time is the time in which the system achieves steady state. System shouldn't overshoot, such that it doesn't violate any of the constraints applied to system.

Fig. 2. shows a block diagram of a general feedback control system. Target system is the system which is being monitored, manipulated and controlled. Transducer is the component which converts the measured output to make it able to be given as input. Transducers are separate components for physical or analog systems which converts to electrical signals readable by controller. Whereas, in computing the output of system is generally transduced. Controller deploys a control law which calculates actuations (control input) for the system. Input to controller is based on reference input, which is target value for system and transduced output which defines the current state of system. Disturbance and noise inputs are those inputs which are not part of the feedback loop in ideal case, but they yet exist and affect target system.

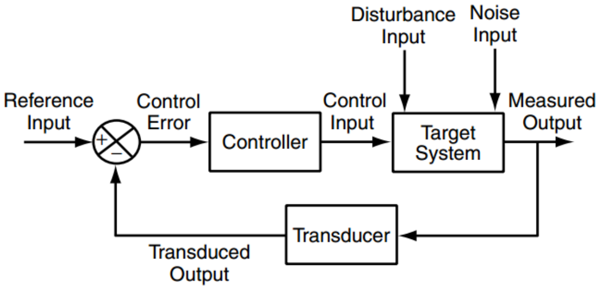

Fig. 2. Block Diagram of a Feedback Control System [18]

Controller in feedback control system is modeled on some control law. A transfer function mathematically defines mapping between input and output of system.

IV. SLA MANAGEMENT WITH CONTROL THEORY

SLA management in cloud is a very crucial process in terms of business value. It has to be a balancing, to avoid SLA violation by not meeting user demands and at same prevent resources being underutilized. It also needs to be accurate and to have timely decision with least human supervision. In this paper we propose a framework which utilizes control theory in SLA management for cloud computing environment. It is also applicable to other application specific SLA management. Feedback control is a continuous process of measuring, monitoring and manipulating a system. It is accurate well modeled problem, timely decision due to its continuing nature and involves least human supervision, near to none. We utilize variation of proportional integral derivative (PID) controller for various SLA metrics. Reference [19] categorizes SLA metrics into metrics for hardware, software, network and storage. It further identifies availability, service times as most important metrics. Reference [20] enlists top one hundred IT performance metrics, which include, response time, solution times (mean time to resolve). These and remaining metrics selected are more of end user interest. The selected SLA metrics are:

Name	***Service Object***	***Unit***
Availability	Hardware, Software, Network, Storage	Time, Percent
Service Times	Hardware, Software, Network	Time Interval
Latency Time	Network	Duration
Response Time	Hardware, Software, Storage	Duration
Number of workstations or licenses	Hardware, Software	Number
Backup Time	Hardware, Storage	Duration
Solution Times	Software, Network	Duration
Failure Frequency	Hardware, Storage	Number
Memory Size	Storage	Number
Accessibility in case of problem	Hardware, Storage	Boolean

TABLE I. Common SLA metrics

Common SLA metrics as shown in table I. have multiple things in common, most importantly measurement unit. Most of these metrics exist for some of four categories. Unit is important in the context as we have to form a mapping between output measured and the input for the system. As in a control system theory the transduced input should have same unit as of reference input. We define following abstract variables associated with each SLA metric:

Name	***Output***	***Input***
Availability	Time ***x*** or percent ***x*** resource ***a*** is available	Number (or list) of resources (including resource ***a***) should be available through service times for time or percent ***x***
Service Times	Resource ***a*** will be in service within time Interval ***x***	Number (or list) of resources (including resource ***a***) should be available through service times for time or percent ***x***
Latency Time	Network ***c*** has latency time ***x***	Latency Check (test case results)
Response Time	Duration ***x*** is taken by resource ***a*** to respond	***x*** number of Virtual Machines (VMs) (providing optimal response time)
Number of workstations or licenses	Service ***b*** is composed of ***x*** number of VMs or number ***x*** of software licenses	Difference between number of SLA and actual VMs
Backup Time	Resource ***a*** is provided backup in duration ***x*** on failure	Location of (nearest and less congested) backup server
Solution Times	Problem occurred with resource ***a*** is solved in duration ***x*** on failure	Solution Time (last and minimum) to optimize
Failure Frequency	Resource ***a*** or service ***b*** has failure frequency ***x***	Number of copies of resource/service such that failure frequency approaches zero
Memory Size	VM ***a*** is allocated ***x*** bytes of memory throughout its life cycle	***x*** number of bytes of memory such that it approaches reference memory size
Accessibility in case of problem	Resource ***a*** or service ***b*** is accessible in case of some problem occurred	Accessibility Check (test case results)

TABLE II. Variables for SLA Metrics

By mapping generic SLA framework on general feedback control system we form a SLA management system based on feedback control system as shown in Fig. 3. With the help of abstract variables given in table II. we define a strategy to measure those metrics from clouds internal measurement service or component provided (measurement and monitoring component) and define controllers (change management component) for metrics.

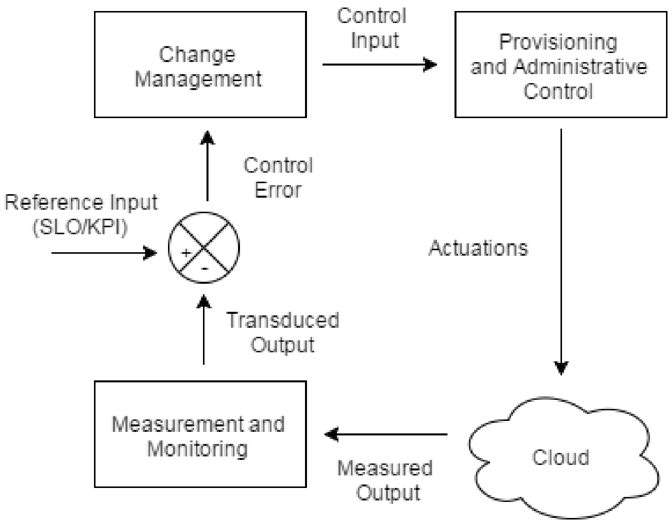

Fig. 3. SLA Management using Feedback Control

Formal Model

Here we define a formal model, a approach to enforce SLA metrics using feedback control. We select a metric which is measured in percentage and the input to system for manipulating that metric is number of resources. The most important metric of all availability is discussed in detail to show the application of feedback control in cloud computing SLA management. Service availability is dependent on the resources which build service. Resources may vary depending on use case i.e. for network resources can be routers, for infrastructure it can be VMs. Here availability is the availability of certain service with respect to its resources at any instance of time. Availability at maximum can be 100% or less.

Definition 1: Desired percentage of availability X_a is percentage of availability given by terms defined by SLO in SLA e.g. preferable availability percentages lie between 90 and 100. In this framework $1 \le X_a \le 100$.

Definition 2: Measured percentage of availability $Y_a(t)$ is the percent availability of resources in cloud against certain resource, we express it as

$$Y_a(t) = \frac{R_a(t)}{R_{SLA}} \times 100 \tag{1}$$

where $R_a(t)$ is number of resources available at an instance of time t and R_{SLA} is desired number of resources to be available as defined by SLA.

Definition 3: Control error $e_a(t)$ for availability metric is error which exist between desired availability and measured percentage of availability, we express it as

$$e_a(t) = X_a - Y_a(t) \tag{2}$$

Definition 4: Cascaded controller gain K_a is a tuning parameter for availability metric, it is common among all (proportional, integral and derivative) control components of PID controller, we express it as

$$K_a = \frac{1}{2\alpha} \tag{3}$$

for fast response (short settling time) α is small and for slow response (long settling time) it is large.

Definition 5: A discrete cascaded PID controller [21] is based on three components. It has common gain K_a for all components. We define transfer function for availability metric as

$$PID_a(t) = K_a e_a(t) + \frac{K_a}{\tau_I}\int_0^t e_a(t)dt + K_a \tau_D \frac{de_w(t)}{dt} \tag{4}$$

where integral time τ_I, derivative time τ_D, sampling time τ_s and with discrete approximations as $\frac{de_a(t)}{dt} = \frac{e_a(t) - e_a(t-1)}{\tau_s}$ and $\int_0^t e_a(t)dt = \tau_s \sum_{i=0}^{t} e_a(i)$.

Definition 6: Required change in availability at time t to fulfill availability requirement is defined as

$$D_{availability}(t) = PID_a(t) \tag{5}$$

Definition 7: Change in number of resources at time t denoted by $R_{new}(t)$ is defined as

$$R_{new}(t) = \left\lceil \frac{PID_{out}(t)}{100} \times R_{SLA} \right\rceil \tag{6}$$

Architecture

Architecture for SLA management and enforcement system to apply this model proposed and used is expressed as in Fig. 4. There are three interconnected parts of architecture.

a) SLA Repository

SLA repository handles contains the SLAs being managed and enforced.

b) SLA Resources

SLA resources represent underlying resources which are being mapped to provide service to service user.

c) SLA Enforcement Component

SLA enforcement component in this architecture is selectable

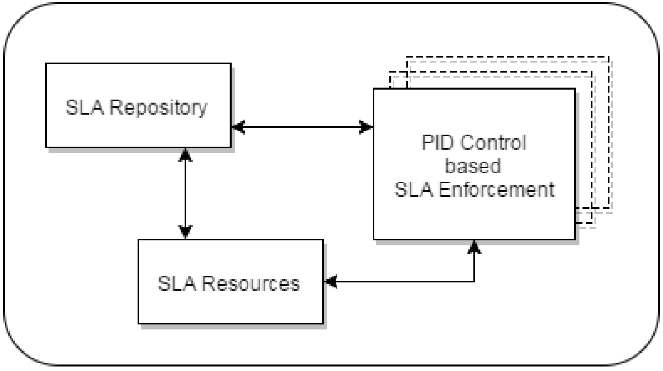

Fig. 4. SLA management and enforcement architecture.

and this architecture can use other techniques in place of PID control as presented in this paper.

Furthermore the architecture can be represented as model shown in Fig. 5. Where SLA Parser and SLA Manager maintain SLA Repository where one parses SLA represented using a standard and other creates a SLA. SLA enforcement is being performed using PID control. Authenticator allows only valid requests to move forward. Cloud Integrator is a interface between SLA resources in this case OpenStack which provides APIs to extend system.

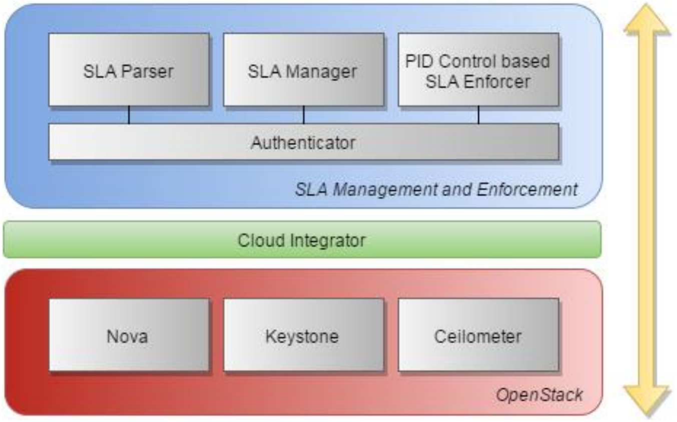

Fig. 5 SLA management and enforcement with OpenStack

V. Results

We simulated the behavior of PID Controller for given conditions including; availability at a timing cannot increase 100% and other varying parameters are expressed in captions. Simulation is performed on the scenario where percentage availability is 60% initially. Figures 6, 7, 8 and 9 show results of simulations for availability metric result is expressed as time plot of PID controller. Figure 6 shows a stable behavior against given parameters for reaching desired percentage of availability. Varying behavior on changing parameters can be seen. The percentage at certain instance of time can be converted to number of resources to make available at certain

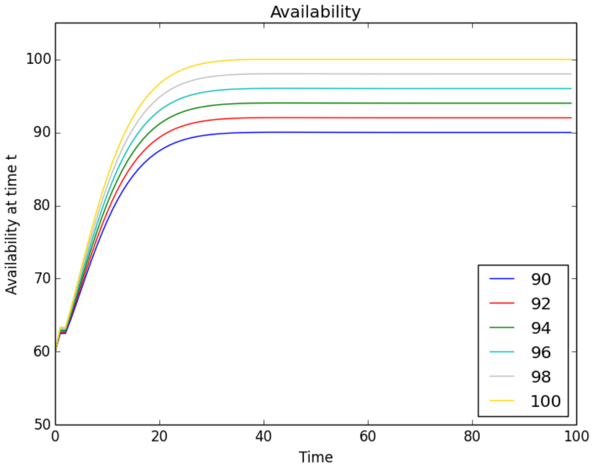

Fig. 6. Simulation results for parameters, initial value = 60%, $\alpha = 17$, $\tau_I = 1.2$, $\tau_D = 1$ and $\tau_s = 0.75$.

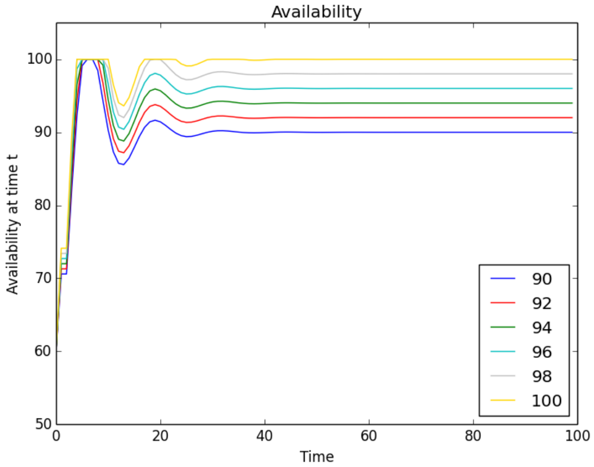

Fig. 7. Simulation results for parameters, initial value = 60%, $\alpha = 17$, $\tau_I = 0.1$, $\tau_D = 1$ and $\tau_s = 0.75$.

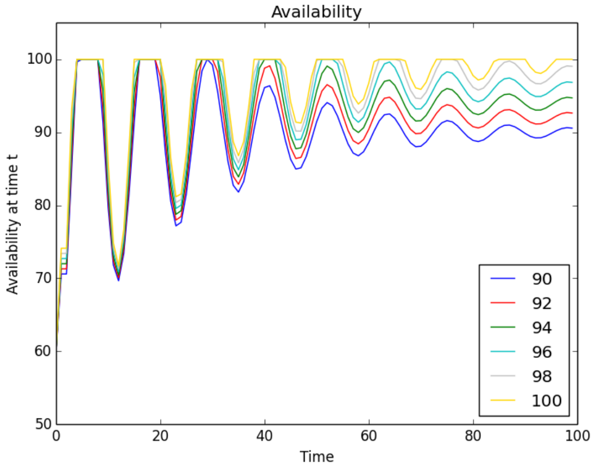

Fig. 8. Simulation results for parameters, initial value = 60%, $\alpha = 17$, $\tau_I = 0.1$, $\tau_D = 1$ and $\tau_s = 0.95$.

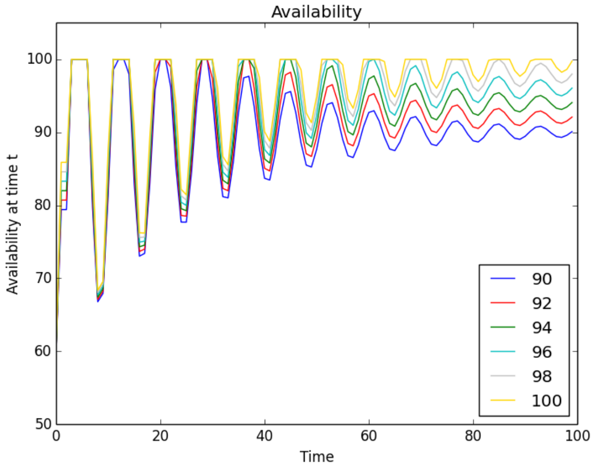

Fig. 9. Simulation results for parameters, initial value = 60%, $\alpha = 17$, $\tau_I = 0.05$, $\tau_D = 1$ and $\tau_s = 0.95$.

time as shown in equation (6). In case of SLA violation this approach will enforce SLA by settling the output using control theory

VI. Conclusion

Cloud computing provides different kinds of service and those services are based on a set of service objects or resources. It requires an automated approach to enforce SLAs such that it fulfills user requirements as specified and prevents over provisioning. It avoids SLA violations by continuously monitoring resources. In this paper we presented a control theory based generic framework for SLA management targeted on SLA enforcement. It utilizes discrete PID controller to enforce different SLA metrics. It is applicable to all kinds of use cases where inputs, outputs of system are clearly identified and categorized. The set of SLA metrics defined with inputs and outputs of system can be utilized as defining place for such systems. Cloud centered SLA metrics can be mapped to this approach. SLA metrics which involve human supervision for enforcement require more information than input and output. Future work involves identification more SLA metrics which can be enforced using control theory, exploration of more control theory approaches and other suitable approaches for SLA management.

References

[1] Andrieux, Alain, et al. "Web services agreement specification (WS-Agreement)." Open Grid Forum. Vol. 128. 2007.

[2] Paschke, Adrian, Jens Dietrich, and Karsten Kuhla. "A logic based sla management framework." Iswc'05: Proceedings of the semantic web and policy workshop. 2005.

[3] Ward, Christopher, et al. "Fresco: a Web services based framework for configuring extensible SLA management systems." Web Services, 2005. ICWS 2005. Proceedings. 2005 IEEE International Conference on. IEEE, 2005.

[4] Zhao, Lu, Salam Sakr, and An Liu. "A framework for consumer-centric SLA management of cloud-hosted databases." (2013).

[5] Lymberopoulos, Leonidas, Emil Lupu, and Morris Sloman. "An adaptive policy-based framework for network services management." Journal of Network and systems Management 11.3 (2003): 277-303.

[6] Comuzzi, Marco, et al. "A framework for multi-level sla management." Service-Oriented Computing. ICSOC/ServiceWave 2009 Workshops. Springer Berlin Heidelberg, 2010.

[7] Emeakaroha, Vincent C., et al. "DeSVi: an architecture for detecting SLA violations in cloud computing infrastructures." Proceedings of the 2nd international ICST conference on Cloud computing (CloudComp'10). 2010.

[8] Brandic, Ivona, et al. "Laysi: A layered approach for sla-violation propagation in self-manageable cloud infrastructures." Computer Software and Applications Conference Workshops (COMPSACW), 2010 IEEE 34th Annual. IEEE, 2010.

[9] Ul Haq, Irfan, Altaf Huqqani, and Erich Schikuta. "Aggregating hierarchical service level agreements in business value networks." Business Process Management. Springer Berlin Heidelberg, 2009. 176-192.

[10] Haq, Irfan Ul, Ivona Brandic, and Erich Schikuta. "Sla validation in layered cloud infrastructures." Economics of Grids, Clouds, Systems, and Services. Springer Berlin Heidelberg, 2010. 153-164.

[11] Ul Haq, Irfan, and Erich Schikuta. "Aggregation patterns of service level agreements." Proceedings of the 8th International Conference on Frontiers of Information Technology. ACM, 2010.

[12] Haq, Irfan Ul, Altaf Ahmad Huqqani, and Erich Schikuta. "Hierarchical aggregation of service level agreements." Data & Knowledge Engineering 70.5 (2011): 435-447.

[13] Appleby, Karen, et al. "Oceano-SLA based management of a computing utility." Integrated Network Management Proceedings, 2001 IEEE/IFIP International Symposium on. IEEE, 2001.

[14] Appleby K., Goldszmidt G., and Steinder M., " Yemanja – A Layered Event Correlation Engine for Multi-domain Server Farms", Proceedings of the Seventh IFIP/IEEE International Symposium on Integrated Network Management, 2001.

[15] Mach, Werner, Benedikt Pittl, and Erich Schikuta. "A Forecasting and Decision Model for Successful Service Negotiation." Services Computing (SCC), 2014 IEEE International Conference on. IEEE, 2014.

[16] Abdelzaher, Tarek, Yixin Diao, Joseph L. Hellerstein, Chenyang Lu, and Xiaoyun ZhuZ. Introduction to Control Theory And Its Application. 2008.

[17] Hellerstein, Joseph. Feedback Control of Computing Systems. New York: IEEE, 2004. 3-7.

[18] Hellerstein, Joseph L., Yixin Diao, Sujay Parekh, and Dawn M. Tilbury. Block Diagram of a Feedback Control System.Feedback Control of Computing Systems. New York: IEEE, 2004. 3-7

[19] Paschke, Adrian, and Elisabeth Schnappinger-Gerull. "A Categorization Scheme for SLA Metrics." Service Oriented Electronic Commerce 80 (2006): 25-40.

[20] Spanos, Nicholas. "100 IT Performance Metrics." 100 IT Performance Metrics. Computer Aid, Inc., Web. 17 Sept. 2015.

[21] Tham, M. "Discretised PID Controllers." Discrete PID Controllers. Chemical Engineering and Advanced Materials, University of Newcastle Upon Tyne, Web. 17 Sept. 2015.