An Approach to Identify and Monitor SLA Parameters for Storage-as-a-Service Cloud Delivery Model

Nirnay Ghosh
School of Information Technology
Indian Institute of Technology
Kharagpur - 721302
Email: nirnay.ghosh@iitkgp.ac.in

Soumya K Ghosh
School of Information Technology
Indian Institute of Technology
Kharagpur - 721302
Email: skg@iitkgp.ac.in

***Abstract*—Cloud computing is an emerging technological paradigm that follows *pay-as-you-use* model. By this model, the consumers are charged according to the usage without regard to where the services are hosted or how they are delivered. Usually cloud services are offered as various delivery models among which *Storage-as-a-Service* cloud is gaining interest in recent time. In this model, raw (block) storage is offered as a service. Like any other utility services, cloud negotiates its service levels and guarantees with its consumers by establishing *Service Level Agreements (SLAs)*. *SLA* is important in any provider-consumer interaction as it defines a formal basis for performance and availability the provider guarantees to deliver. It consists of a set of measurable attributes called *SLA parameters* which are established by some objectively measurable conditions, termed as *Service Level Objectives (SLOs)*. However, the *SLA*s provided by the present day state-of-the-art clouds are relatively biased towards vendors and do not provide any formal method of verifying if the guarantees are complying or not. This paper attempts to identify the *SLA parameters* for *Storage-as-a-Service* cloud delivery model and proposes a monitoring framework for compliance checking.**

I. Introduction

Service Level Agreement (SLA) is a formal contract between a *service provider (SP)* and a *service consumer (SC)* and defines performance and availability levels which the *SP* guarantees to deliver. It documents common understandings about agreement features such as priorities, responsibilities, and guarantees. The objective of an *SLA* is to reduce area of potential conflicts and their resolution before these issues get materialized. It can be used to communicate and coordinate performances. Different vendors offer different *SLA* structures, service offerings, performance levels, and negotiation opportunities. Therefore, *SLAs* can also be used to select vendors on the basis of data protection, continuity, and cost [1]. A typical *SLA* will contain the following [2] : (i) a set of services which the provider will deliver, (ii) a complete, specific definition of each service, (iii) the responsibilities of the provider and the consumer, (iv) a set of metrics to measure whether the provider is offering the services as guaranteed, (v) exclusion clauses, (vi) an auditing mechanism to monitor the services, (vii) the remedies available to the consumer and the provider if the terms are not satisfied, and (ix) how the *SLA*s will change over time. The present cloud marketplace features the following *two* types of *SLA*s [3] :

1) *Off-the-shelf SLA or non-negotiable SLA/Direct negotiation*: Not conducive for mission-critical data or applications. Provider creates an unique template and defines all *SLA* criteria such as period of contract, billing, and response time.
2) *Negotiable SLA*: Terms are expensive and has two categories:
   - *Negotiation via trusted agent*: A trusted external agent selects the cloud provider and defines the critical parameters for *SLA*.
   - *Negotiation via multiple trusted agent*: Efficient when the cloud consumer requires more than one type of cloud services.

The service qualities which the *SP* guarantees to offer through the *SLA* are measured by some metrics based on which monitoring and auditing of *SLA*s may be done. These metrics are known as *SLA parameters*. Some common examples of *SLA parameters* are *throughput, availability, reliability, response time, service-level violation rate, transaction time, resolution time*, and so on. The *SLA parameter* used by the present day cloud providers is mostly restricted to *availability* of the service. However, there exists other non-trivial parameters specific to each cloud delivery models whose inclusion is necessary to render completeness to the *SLA*. This is because, the consumers not only demand service availability guarantees, but also other performance and management related assurances for business process continuity. *SLA parameters* are usually established by defining some objectively measurable conditions which are termed as *service level objectives (SLOs)*. *SLO*s are specific measurable characteristics of the *SLA parameters* and represent the level of quality-of-service which is expected to be provided. Some examples of *SLOs* are as follows:

- ***Availability*** of a service X is 99.5
- ***Response time*** of a database query Q is between 3 to 5 seconds

978-1-4673-4941-3/12/$31.00 ©2012 IEEE

- ***Throughput*** of a server S at peak load time is 0.875

Measurement of *SLA* guarantees with respect to the originally negotiated ones is done by *SLA parameters*. Each high-level *SLA parameter* is a function of one or more *key performance indicators (KPIs)* [4] [5], which are low-level resource metrics that are composed, aggregated, or converted to form high-level *SLA* parameters. A possible mapping between a high-level *SLA parameter* to low-level *KPI*s [5] is given in Table II.

Table I
MAPPING BETWEEN SLA PARAMETER AND KPIS

SLA Parameter	**KPI**	**Mapping**
Availibity	uptime, downtime	Availability = 1 - $\frac{downtime}{uptime}$
Response Time (R_{total})	inbytes, outbytes, packetsize avail.bandwidthin, avail.bandwidthout	R_{total} = R_{in} + R_{out}, R_{in} = $\frac{packetsize}{avail.bandwidthin-inbytes}$, R_{out} = $\frac{packetsize}{avail.bandwidthout-outbytes}$

In this paper, formal representation of these *SLOs* pertaining to *Storage-as-a-Service* cloud delivery model has been attempted. Formalization is necessary for : (i) standardization of *SLA parameters* to determine service levels and guarantees to be offered, and (ii) verifying if the guarantees are in compliance with the negotiated ones.

The rest of the paper is organized as follows: Section II presents the motivation and objectives of the work. In Section III, identification of *SLA parameters* for *Storage-as-a-Service* cloud has been done, followed by the formal representation of *Service Level Objectives*. Section IV proposes a novel *SLA* monitoring framework for compliance checking. Finally, section V draws conclusion by giving directions to the future work.

II. MOTIVATION & OBJECTIVES

Cloud computing is conceived to be the future of the present day business model in which organizations will host their data and applications without regard to where and how they are hosted. Therefore, it is essential to have trust relationship between the provider and the consumer as the later does not have complete control over his application in the cloud environment. If there exists a proper and unbiased *SLA*, it is beneficial to both the parties in the following manner:

- *Providers' perspective* : Improve upon cloud infrastructure and maintain a fair competition among themselves.
- *Consumers' perspective* : Choosing appropriate provider based on the quality of service guaranteed.

However, the present day state-of-the-art cloud provides the *SLA*s which have the following limitations:

1) *Service measurement* : It is restricted to service uptime percentage and is measured by taking the mean availability observed over a specific time interval. It ignores other parameters like stability, capacity, load balancing, etc.
2) *Biasness towards vendors* : Measurement of parameters are mostly established according to vendor's advantage.
3) *Lack of active monitoring on the consumer's side* : Consumers are given access to some ticketing systems and are responsible for monitoring the outages. Providers do not provide any access to active data streams or audit trails, nor do they report any outages.
4) Gap between *QoS hype* and *SLA offerings* in reality.
5) QoS in the areas of *governance*, *reliability*, *availability*, *security*, and *scalability* are not well addressed.
6) Lack of formal ways to verify if the *SLA* guarantees are complying or not.

The *SLAs* offered by cloud providers like *Amazon*, *Google*, *Microsoft*, *Salesforce.com*, etc. do not render the consumers with the ability to monitor and measure the performances based on *Service Level Objectives (SLOs)*. The *SLA parameter* is also restricted mostly to *service availability* but offers *service credits* if there is a violation of negotiated guarantees. However, in most cases, it is the responsibility of the consumers to detect any service outage. An overview of the *SLAs* provided by different vendors (obtained mostly from public domains) has been summarized in Table I. It can be observed that the "availability" is prevalent in all Cloud *SLA*s, exception being *Rackspace* which offers performance and recovery time guarantees. However, other than "availability" there exists other *SLA parameters* specific to each cloud delivery models whose inclusion is necessary to render completeness to any *SLA*. This is because, consumers not only demand availability guarantee but also other performance related assurances which are extremely business critical [18]. **Hence, there is a need to identify non-trivial *SLA parameters* for different cloud delivery models, to facilitate monitoring and auditing of cloud performances from various perspectives**. Considering the above mentioned limitations of the state-of-the-art cloud *SLA*s, the **objectives** of the current work can be summarized as follows:

- Identifying non-trivial *SLA parameters* for *Storage-as-a-Service* cloud which are not offered by the present day cloud vendors.
- Proposing a novel *SLA* monitoring framework to facilitate compliance checking of *SLOs* by a trusted third party.

III. IDENTIFYING SLA PARAMETERS FOR STORAGE-AS-A-SERVICE CLOUD DELIVERY MODEL

In *Storage-as-a-Service* cloud delivery model, raw (block) storage is offered as a service. Rather than purchasing physical storage devices, like disks, tapes, etc., clients instead utilize them as fully outsourced service. Some of the leading storage providers in the present cloud marketplace are *Nirvanix*, *Dropbox*, *Microsoft Azure Storage*, and so on. In Section II, a vivid description of the *SLA* offered by different providers has been presented. Majority of the storage providers use *availability* as the major *SLA parameter*. However, other than *availability*, a number of non-trivial *SLA parameters*, which are measurable, can be included as part of *SLA*. In this section, *SLA parameters* relevant to *Storage-as-a-Service* cloud and

Table II
SLA GUARANTEES OFFERED BY DIFFERENT PROVIDERS

Cloud Provider	Service	Delivery Model	Service Level Agreement Guarantees
Amazon [6]	EC2 [7]	Infrastructure-as-a-Service (IaaS)	Availability (99.95%) with the following definitions: Service Year: 365 days of the year, Annual Percentage Uptime, Region Unavailability: no external connectivity during a five minute period, Eligible Credit Period, Service Credit
	S3 [8]	Storage-as-a-Service	Availability (99.9%) with the following definitions: Error Rate, Monthly Uptime Percentage, Service Credit
	SimpleDB [9]	Database-as-a-Service	No specific SLA is defined and the agreement does not guarantee availability
Salesforce [10]	CRM	Platform-as-a-Service (PaaS)	No SLA guarantees available in the public domain
Google [11]	Google App Engine	Platform-as-a-Service (PaaS)	Availability (99.9%) with the following definitions : Error Rate, Error Request, Monthly Uptime Percentage, Scheduled Maintenance, Service Credits, and SLA exclusions
Microsoft	Microsoft Azure Compute [12]	IaaS/PaaS	Availability (99.95%) with the following definitions : Monthly Connectivity Uptime Service Level, Monthly Role Instance Uptime Service Level, Service Credits, and SLA exclusions
	Microsoft Azure Storage [13]	Storage-as-a-Service	Availability (99.9%) with the following definitions: Error Rate, Monthly Uptime Percentage, Total Storage Transactions, Failed Storage Transactions, Service Credit, and SLA exclusions
Zoho suite [14]	Zoho mail, Zoho CRM, Zoho books	Software-as-a-Service (SaaS)	Allows the user to customize the service level agreement guarantees based on : Resolution Time, Business Hours & Support Plans, and Escalation
Rackspace [15]	Cloud Server	IaaS	Availability regarding the following: Internal Network (100%), Data Center Infrastructure (100%), Load balancers (99.9%) Performance related to service degradation: Server migration, notified 24 hours in advance, and is completed in 3 hours (maximum) Recovery Time: In case of failure, guarantee of restoration/recovery in 1 hour after the problem is identified.
Terremark [16]	vCloud Express	IaaS	Monthly Uptime Percentage (100%) with the following definitions: Service Credit, Credit Request and Payment Procedure, and SLA exclusions
Nirvanix [17]	Public, Private, Hybrid Cloud Storage	Storage-as-a-Service	Monthly Availability Percentage (99.9%) with the following definitions: Service Availability, Service Credits, Data Replication Policy, Credit Request Procedure, and SLA Exclusions.

the related low-level *key performance indicators* have been identified and mapped to render completeness to the *service level agreement*.

A. SLA Parameters for Storage-as-a-Service Cloud

Instrumenting *SLA parameters* for *storage-as-a-service* cloud requires considering not only performance or availability issues, but also the degree of fault tolerance the cloud offers during disasters. As consumers store sensitive and business data onto the cloud, it is mandatory to maintain certain level of redundancy to ensure operational continuity. Beside this, *SLA* should include security and access control issues to guarantee secured data transaction to and from the cloud. Some critical management issues, such as, disaster recovery, data life cycle management, and so on, should be perceived as part of storage cloud *SLA*. As cloud data centers may be placed at geographically dispersed locations having different regulatory policies, the providers are bound to abide by these regulations to ensure confidentiality of the consumer data. Therefore, the *SLA* should contain a mechanism for checking compliances of these regulatory policies.

Considering the above arguments, other non-trivial *SLA parameters* for the storage cloud can be proposed as: (i) Fault Tolerance, (ii) Performance, (iii) Disaster Recovery, (iv) Security, (v) Governance, (vi) Data Life Cycle Management (DLM), (vii) Error Rate.

The associated *key performance indicators (KPIs)* for each high-level *SLA parameters* have been identified in Table III.

B. Formalization of Service Level Objectives (SLOs)

As discussed in Section I, *Service Level Objectives (SLOs)* are specific measurable characteristics of the *SLA parameters* and represent the level of quality-of-service which is expected to be provided. For storage cloud, the *SLOs* should be comprised of the *SLA parameters* proposed in section III-A. In this subsection, formalization of *SLOs* in *first-order predicate logic* has been attempted to facilitate compliance checking through logic solvers. Formalization of *SLOs* pertaining to the parameter *performance* is presented in Table IV. Some of the *SLOs* are of generic nature which can be instantiated depend-

Table III
SLA PARAMETERS AND CORRESPONDING KPIS

SLA Parameter	Key Performance Indicators (KPIs)
Fault Tolerance	**Data Replication**: different categories include synchronous, asynchronous, semi-synchronous, and point-in-time **Data Mirroring** **Multipath IO**
Performance	**Type of Application**: transaction processing, scientific application, etc. **Maximum number of User Requests** **Response Time**: Read IO Latency & Write IO Latency **Transferring Bandwidth**: inbytes, outbytes, and degree of link redundancy
Disaster Recovery	**Recovery Point Objective (RPO)**:The maximum amount of data that will be lost following an interruption or disaster. **Recovery Time Objective (RTO)**: The period of time allowed for recovery i.e., the time that is allowed to elapse between the disaster and the activation of the secondary site.
Security	**Confidentiality**: The storage cloud should be able to isolate data in a multi-tenant environment to support confidentiality or privacy as well as industry standard identity management. **Integrity**: Data should be prevented from tampering to restore integrity. This demand for data protection against worms, viruses, spywares, Trojans, or even scripting, application-specific, and injection attacks. **Availability**: The storage service and data should be available whenever the consumer requires them. This requires prevention of *Denial-of-Service (Dos)* attacks which includes *SYN flooding*, *ICMP flooding*, etc. **Authentication**: Determining whether the access rights on the data in the storage cloud are in conformance with the privilege levels. This tenet also encompasses auditing and non-repudiation mechanisms. **Authorization**: Involves verifying that an authenticated subject has permission to perform certain operations on access specific resources.
Data Life Cycle Management (DLM)	**Data Archival**: Determines whether the data has been archived from Tier-I storage to tapes only after a predefined period of time. **Accessibility of the Archived Data**: Determines the privilege level of the consumer who accesses archived data. **Access Time**: For newer data that needs to be accessed frequently, latency should be lower as the data is fetched from faster and expensive storage media. For obsolete data which has been archived the latency may be high depending on the type of the storage media where it is archived.
Governance	**Geographic Location**: Different geographically dispersed locations where the cloud provides storage service. **Regulations**: Regulatory compliance policies that influence the data storage at various geographic locations. **Availability**: Determines if a data co-located at geographically dispersed locations are available, if one of the sites goes down.
Error Rate	**Total Storage Transactions**: All the storage transactions in a given time interval (initially set at one hour) for a subscription, with a few notable exceptions. **Failed Storage Transactions**: Includes those transactions which fail to complete within a predefined processing time.

ing upon the type of implementation. The primitives used for formal representation of *SLOs* relevant to *performance* have been defined in Table V.

Table V
PRIMITIVES

Primitives	Definition
$job(x)$	Any job identifier x
$performance\_of(x)$	Performance of job x
$request(r)$	Any request identifier r
$response\_time(x)$	Response time of job x
$read\_IO\_latency(x)$	Read latency corresponding to job x
$write\_IO\_latency(x)$	Read latency corresponding to job x
$RAID\_level(p)$	Any RAID level identifier p
$link\_availability(y)$	Availability of any link y
$data\_transfer\_bw(x)$	Data transfer bandwidth relevant to job x
$inbytes(x)$	Amount of data written into cloud storage corresponding to job x
$outbytes(x)$	Amount of data written into cloud storage corresponding to job x

IV. CLOUD SLA MONITORING FRAMEWORK

In Section II, the limitation of the present day state-of-the-art cloud *SLAs* has been mentioned. One of the major drawbacks is the inability of the consumers to monitor and audit the performances of the cloud services. Different monitoring schemes, based on the responsibilities of the service provider or the consumer, have been proposed in the literature. In the present work, a monitoring model, influenced by the features of *periodic probing with trusted probe clients* [19], has been conceived. Below are the assumptions for the proposed monitoring model :

- The *SLA* guarantees offered by the provider are disclosed to a trusted third-party negotiator-cum-auditor.
- The monitoring and auditing mechanisms are performed by the trusted third-party arbitrator.
- The arbitrator may perform *SLA* guarantee compliance checking at any point of time on request from either the consumer or the provider.
- If there is a non-compliance, the penalties to be credited are mutually settled down by the contracting parties.

The proposed *SLA* monitoring architecture is depicted in Figure 1. It consists of three participating entities : (i) Provider,

Table IV

FORMALIZATION OF SLOS FOR PERFORMANCE

Sl. No.	Service Level Objectives & Logic Representation
1.	For all jobs, there is a maximum number of requests which the cloud can handle without degrading its performance at any given point of time $\forall x \forall t \forall r[job(x) \wedge time(t) \wedge request(r) \wedge (r \leq max\_user\_request\_processed)]$
2.	For all jobs, the performance of the cloud is measured by the response time and the data transferring bandwidth $\forall x[job(x) \wedge performance\_of(x) \Rightarrow response\_time(x) \wedge data\_transfer\_bw(x)]$
3.	For any job, the response time is measured by the read and write IO latencies $\forall x[job(x) \wedge response\_time(x) \Rightarrow read\_IO\_latency(x) \wedge write\_IO\_latency(x)]$
4.	For all job, the read IO latency depends upon the type of RAID technology used and is measured in terms of the seek time $\forall x[job(x) \wedge read\_IO\_latency(x) \Rightarrow \forall p[RAID\_level(p) \wedge seek\_time(p)]]$
5.	For all jobs, the write IO latency depends upon the type of RAID technology used and is measured in terms of the seek time $\forall x[job(x) \wedge write\_IO\_latency(x) \Rightarrow \forall p[RAID\_level(p) \wedge seek\_time(p)]]$
6.	For all jobs, the data transferring bandwidth is the amount of inbytes and outbytes of data the links can support and the link availability depends on the degree of redundancy $\forall x \exists y[job(x) \wedge link\_available(y) \wedge data\_transfer\_bw(x) \Rightarrow inbytes(x) \wedge outbytes(x)]$
7.	For all jobs at all time, the amount of inbytes supported should be greater than a predefined threshold value and can extend up to the maximum allowable incoming bandwidth given that at least one link is in active state $\forall x \forall t \exists y[job(x) \wedge time(t) \wedge link\_available(y) \wedge (threshold\_inbytes < inbytes(x) \leq max\_available\_in\_bw)]$
8.	For all jobs at all time, the amount of outbytes supported should be greater than a predefined threshold value and can extend up to the maximum allowable outgoing bandwidth $\forall x \forall t \exists y[job(x) \wedge time(t) \wedge link\_available(y) \wedge (threshold\_outbytes < outbytes(x) \leq max\_available\_out\_bw)]$

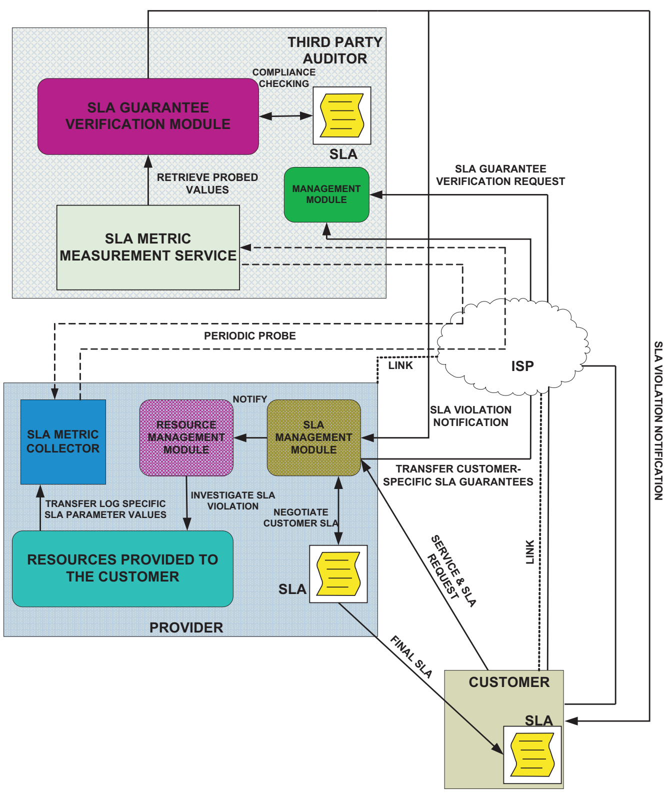

Figure 1. Cloud SLA Monitoring Architecture

(ii) Consumer, and (iii) Third-party auditor. However, the *Internet Service Provider (ISP)* can also be considered to be a participating entity, as in almost all cases, performance of cloud services is dependent on the *bandwidth* and *QoS* provided by the *ISP*. Various modules constituting the *provider* entity are as follows:

1) *SLA Management Module* : This module negotiates consumer *SLA*s with those of the provider and transfers the mutually settled *SLA* guarantees to the auditor and receives *SLA* violation notification from the later.
2) *Resource Management Module* : On receiving *SLA* guarantee violation notification, it investigates which physical resource is responsible for this infringement.
3) *SLA Metric Collector* : Captures application and system logs corresponding to the *key performance indicators (KPIs)* constituting the *SLA parameters* as and when requested by the auditor.

Different modules constituting the *third-party auditor* are given as:

1) *Management Module* : It manages the *SLA* guarantee verification request from different consumers.
2) *SLA Metric Measurement Module* : In this module, the high-level *SLA parameters* are computed from the obtained *KPIs* using the defined mappings among them. The values corresponding to the *KPIs* are obtained by sending periodic probes to the provider site's *SLA Metric Collector*.
3) *SLA Guarantee Verification Module* : This module verifies the compliance of the computed values of the *SLA parameters* with the guaranteed values. If there is a violation, the notifications are send to both the provider and the consumer.

V. CONCLUSION

Cloud computing enables the organizations to exempt themselves from management and maintenance of in-house IT resources and utilize cloud services following *pay-as-you-use* model. Interaction between a cloud service provider and a customer is usually preceded by documenting the guarantees in a formal contract, called *Service Level Agreement (SLA)*. It forms the basis for performance and availability the provider guarantees to deliver. However, the present day state-of-the-art cloud *SLAs* have a number of limitations, particularly, with respect to defining *SLA parameters* and monitoring them for compliance checking of the *Service Level Objectives (SLOs)*. Good *SLA* is necessary from both the provider as well the consumer perspective. In this paper, the authors have attempted to identify different non-trivial *SLA parameters* for *Storage-as-a-Service* cloud delivery model, and then formally represent the *SLOs* in *first-order predicate logic*. Finally, a novel monitoring framework has been proposed to enable trusted third-party auditors to accomplish compliance checking. In future, the identification of non-trivial *SLA parameters* for other cloud delivery models will be undertaken and the formal definitions of *SLOs* will be used to verify negotiated guarantees using a logic solver.

REFERENCES

[1] K. Buck and D. Hanf, "Cloud SLA Consideration for Gevernment Consumer," System Engineering at Mitre, Tech. Rep. Case: 10-2902, September 2010, *Available at http://www.mitre.org/work/tech_papers/2010/10_2902/*.

[2] "Cloud Computing Use Cases White Paper - Version 4.0," July 2010, *Available at : http://opencloudmanifesto.org/Cloud_Computing_Use_Cases_Whitepaper–4_0.pdf*, accessed on August 2011.

[3] M. Alhamad, T. Dillon, and E. Chang, "Conceptual sla framework for cloud computing," in *Proceedings of the 4th IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2010)*, 13-16 April 2010, pp. 606–610, *DOI : 10.1109/DEST.2010.5610586*.

[4] 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," in *Proceedings of the 2010 34th IEEE Annual Computer Software and Applications Conference Workshops (COMPSACW)*, 19–23 July 2010, pp. 365–370, *DOI : 10.1109/COMPSACW.2010.70*.

[5] V. C. Emeakaroha, I. Brandic, M. Maurer, and S. Dustder, "Low level metrics to high level slas - lom2his framework:bridging the gap between monitored metrics and sla parameters in cloud environments," in *Proceedings of International Conference on High-Performance Computing and Simulation (HPSC)*, June 28 – July 2 2010, pp. 48–54, *DOI : 10.1109/HPCS.2010.5547150*.

[6] "Amazon Web Services : Products & Services," 2011, *Available at : http://aws.amazon.com/*, accessed on August 2011.

[7] "Amazon EC2 Service Level Agreement," 2011, *Available at : http://aws.amazon.com/ec2-sla/*, accessed on August, 2011.

[8] "Amazon S3 Service Level Agreement," 2011, *Available at : http://aws.amazon.com/s3-sla/*, accessed on August 2011.

[9] "Amazon SimpleDB," 2011, *Available at : http://aws.amazon.com/simpledb/*, accessed on August 2011.

[10] "The Promise and Pitfalls of Service Level Agreements," 2011, *Available at : http://www.online-crm.com/sla.htm*, accessed on August 2011.

[11] "Google Apps : Google Apps Service Level Agreement," *Available at : http://www.google.com/apps/intl/en/terms/sla.html*, accessed on August 2011.

[12] "Windows Azure Compute Service Level Agreement (SLA)," 2011, *Available at : http://www.microsoft.com/windowsazure/sla/*, accessed on August 2011.

[13] "Windows Azure Storage Service Level Agreement," 2011, *Available at : http://www.microsoft.com/windowsazure/sla/*, accessed on August 2011.

[14] "Zoho Support : Service Level Agreement - SLA," 2010, *Available at : http://www.zoho.com/support/help/sla-settings.html*, accessed on August 2011.

[15] "Rackspace Cloud Legal : Service Level Agreement (SLA)," 2011, *Available at : http://www.rackspace.com/cloud/legal/sla/*, accessed on August 2011.

[16] "Terremark Cloud Service Level Agreement," 2009, *Available at : https://community.vcloudexpress.terremark.com/en-us/product_docs/w/wiki/service-level-agreement.aspx*, accessed on August 2011.

[17] "Nirvanix Service Level Agreement," 2011, *Available at : http://www.nirvanix.com/sla.aspx*, accessed on August 2011.

[18] G. S. Machado and B. Stiller, "Investigation of an sla support system for cloud computing (slacc)," *PIK (Praxis der Informationsverarbeitung und Kommunikation)*, vol. 34, no. 2, pp. 80–86, June 2011, *DOI : 10.1515/piko.2011.014*.

[19] C. Molina-Jimenez, S. Shrivastava, J. Crowcroft, and P. Gevros, "On the monitoring of contractual service level agreements," in *Proceedings of the First International Workshop on Electronic Contracting*, July 6 2004, pp. 1–8, *DOI : 10.1109/WEC.2004.1319502*.