A Dynamic Model for automatic Updating cloud computing SLA (DSLA)

Abdel-Rahman Al-Ghuwairi
Hashemite University
Software Engineering Department
ghuwairi@hu.edu.jo

Mohammad.N.Khalaf
Hashemite University
Software Engineering Department
hamoodnooman@itc.hu.edu.jo

Laith Al-Yasen
Hashemite University
Software Engineering Department
laith.saleh90@yahoo.com

Zaher Salah
Hashemite University
Software Engineering Department
zaher@hu.edu.jo

Ahmad A. Al-oqaily
Hashemite University
Software Engineering Department
aloqaily@hu.edu.jo

Aladdin Hussein Baarah
Hashemite University
Software Engineering Department
aladdin.baarah@hu.edu.jo

ABSTRACT

Nowadays, Cloud computing became one of the most important fields in computer paradigm researches, because it provides many advantages and benefits for all users whether the organizations or the individuals, by granting them the needed resources and services. Service level agreement (SLA) is an essential topic in terms of cloud computing, it is a contractual agreement signed between cloud user and cloud provider in order to manage and define the level of access for the resources and quality of services provided. Continues changes of technologies and requirements for both user and provider make updating SLA necessary, to ensure that services provided are meeting their needs among multiple QoS (Quality of Service) parameters. Intensive studies have been presented for building a flexible SLA. In our study, we proposed a dynamic model for updating cloud computing SLA overtime, in order to get a flexible, adaptive SLA and avoid costly violations.

Keywords

Cloud computing; QoS; SLA

1. INTRODUCTION

Cloud computing is a new trend of IT technology that is composed of multi other technologies such as SOA, Grid computing, virtual machine, utility computing and web service. Cloud computing was introduced in the sixties by John McCarthy as a public utility delivering computation system, then the web 2.0 allowed the cloud computing to evolve over time to reach the way it is today. The lead company of using cloud computing is Salesforce in 1999 then amazon company started to provide cloud computing service in 2002, recently a lot of company enters the competition of providing cloud computing service such as google by google app engine and microsoft by Microsoft azure. Today there are a lot of definitions and less of standards of cloud computing due to the viours views in this industry. The most famous companies that provides cloud computing service are Amazon, Google, Microsoft, dell, HP, Rackspace and apple. Each one of them uses different concept and definition for cloud computing in there product. Recently there are many researchers from academic view try to define a standards for cloud computing such as NIST [12] ,CDMI and OCCI for using it in the cloud industry, to made cloud approach more clear and understandable, researcher start of using formalities and math to propose an understandable approach.

Most recent researches in computing paradigm such as cloud computing, start growing rapidly. Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. [12] Cloud computing means to grant services and resources to cloud user by cloud provider from pool of resources through internet. The main characteristics of cloud computing are : On-demand self-service: cloud user get the service from cloud provider in bases of what he demand. Broad network access: the capabilities of providing a service over a network which can be accessed through various standard machine such as tablets, laptops, PC, and mobile phone etc. Resource pooling: it's a cloud computing term to a cloud provider serve multi-client and customers using a scalable services suitable for all user without any change on them. Rapid elasticity: it's an IT term implies for cloud computing that scalable provisioning, or the capability of cloud computing to provide services that are scalable. Measured service: the services and resource are controlled, monitored and reported to cloud user, this applying by providing visibility for consumption and associated cost.

Cloud computing provides three types of services; software as a service (SaaS)such as Google Apps, Salesforce, Workday, Concur, Citrix GoToMeeting, and Cisco WebEx, platform as a service(PaaS)such as Apprenda, Infrastructure as a service(IaaS)such as Amazon Web Services, Microsoft Azure, Google Compute Engine , and Joyent. From privacy point of view cloud computing have four types of system private cloud, public cloud, community and hybrid cloud. Cloud computing consist of three parties; cloud provider, cloud user and third party(optional). To provide efficient and reliable service we must cover functional requirement and non-functional requirement as well, NFR such as performance, security, response time, availability, reliability etc. in the cloud system, these requirement called QoS.

One of the problems in cloud systems is how to guarantee the quality of service granted by cloud provider to the cloud user. A contract between cloud provider and cloud user has been proposed to manage the relationship, this contract is called SLA. The latest researches proposed many approaches argued that a third party should be responsible for monitoring and managing the SLA activities and violations. SLA acts as a contract to determine the following concerns: what the provider promises, how the provider will deliver on those promises, who will measure the delivery and how, what happens if the provider fails to deliver as promised, and how the SLA will change over time. As technology changes over time, SLA will be out dated or inconvenient with the new technologies. So that, SLA should be flexible and able to be changed, otherwise, it will be useless. Our approach discussed the ability of changing the SLA terms at run time. We attempted to give more flexibility for SLA by proposing model for the changed request, a change may be requested by one of the parties or an automated request generated by analyzing historical data of the SLA. Our proposed model will be discussed in more details in the next sections.

For the rest of the paper, cloud service level agreement is presented in Section 2. proposed model is presented in section 3. An example of proposed model is presented in Section 4. related work is presented in Section 5. Finally, in Section 6, a conclusion and future work are presented.

2. CLOUD SERVICE LEVEL AGREEMENT

In this section we present the defenition of service level agreement and quality of service parameters. A real world senarios are expressed in some details in this section too.

2.1 Service level agreement

Service level agreement starts to spread in the 80s in the e-commerce companies to assure the quality of their services, in the late nineties web provider company started support service level agreement concept to arise confidence among their user. The first biggest evolution was on 2001 by web service level agreement (WSLA) is SLA standard. After that, a lot of cloud provider company was adhered such as amazon and google, each company has a unique SLA approach and mechanism, and recently there are many academic approaches and mechanisms to monitor and manage SLA. SLA serves as the foundation for the expected level of service between the consumer and the providers in [10]. From this definition and the issues discused. Above we need to manage and monitor the relation between cloud user and cloud provider in addition to guarantee the quality of Services that cloud providers provides. All the problems above were solved by Service level agreement. Service level agreement is a contract between cloud user and cloud provider to guarantee and serves the expected level of services which cloud user has. Quality of services which is a part of service level agreement change constantly during service level agreement life cycle [2] during to that we are demanding to create a SLA that change over time to cover these changes in QoS. Even if cloud party that uses cloud system is in the same company, the SLA agreement still useful by increasing the effectiveness of using cloud system which is leading to improve the whole cloud computing system in the company.

2.2 Quality of service (QoS)

The concept of QoS is inheritance from the concept of software quality, to be able to define the attribute of QoS you must customize software quality attribute to fit with the cloud computing environment.The QoS concept firstly was for telephone then transfer for internet services, and like other concepts the cloud inherent this concepts, QoS becomes one of the most critical issues of the cloud company, because it considers the cloud user in addition to cloud provider. A QoS for cloud computing equivalent to non-functional requirement in software engineering, here to produced a good service level agreement you must guarantee the quality of service is exactly or near exactly like the cloud user request. A QoS contains many parameters such as security, performance, safety, availability, reliability, response time, bandwidth. In our approach we consider to chose the most important and measurable QoS in SLA which are performance, availability, reliability and bandwidth.

2.3 Real World Service Level Agreement Scenarios

In this section, we present SLA as an example for the most popular cloud provider in which the monthly uptime% and penalty for each violation value. Google app [1] engine provides a monthly guarantee of the uptime quality of service which guarantee that the QoS would not decrease from 99.9 %, the penalty value will be different from other cloud provider credit amount.

Table 1: SLA of Google App

Monthly Uptime Percentage	Days of Service added to the end of the Service term.
99.9%-99.0%	3 days
99.0%-95.0%	7 days
<95.0%	15 days

Table 1 shows us the levels of uptime and the credit amount of each level. Microsoft Azure:

Cloud Microsoft azure [5] is starting to provide infrastructure as a service because it is a leader to cloud computing

Table 2: SLA of Microsoft azure

Monthly Uptime Percentage	SERVICE CREDIT
99.9%-99.0%	10%
<99.0%	25%

future, The Microsoft azure proposes a credit penalty policy for storage, The Microsoft azure guarantees 99.99% of monthly uptime percentage. If the cloud provider delivers from 99% and less than 99.9% of the service, a 10% of the amount is reduced as a penalty, while 25% of the amount is reduced if the cloud provider delivers less than 90%.

Dell cloud dedicated: Dell cloud dedicated [3] is a virtual machine based on data center private cloud, they provide comprehensive data protection of infrastructure as a service cloud system. The dell cloud dedicated provides a penalty mechanism that guarantees 99.95% of monthly uptime percentage.

Table 3: SLA of Dell cloud dedicated

average monthly up-time percentage	Credit Percentage Amount
100%-99.5%	0%
99.94%-99.95%	10%
99.94%-99.0%	25%
98.99%-98.0%	50%
97.99%-96.50%	75%
<96.50	10%

When the cloud provider delivers more than 99.95 percentage of the service, there will be no penalty, while a penalty of 10 % of the amount is reduced when it provides[99.94-99.95 %], a 25% reduction for [99.0-99.94 %], 50% reduction for [98.0-98.99%] of the service,75% reduction for [96.50-997.99%] of the service but the cloud provider returns 100% of the total amount if the cloud provider delivers less than 96.50%

If the cloud provider delivers from 99.50% and less than 99.94% a 10% of the amount is reduced as a penalty , if the cloud provider delivers less than 99.49% and more than 99% 25% of the service monthly credit amount will be refunded to the cloud user as a penalty, If the cloud provider delivers from 98.00% and less than 98.99% of the service, a 50% of the amount is reduced a penalty while 75% of the amount is reduced if the cloud provider delivers from 96.50% and less than 97.99%,and if the cloud provider delivery is less than 96.59 then the penalty will be 100% of service credit.

Amazon S3/EC2: Amazon EC2/S3 [2] provider is an elastic cloud computing that provides web service to compute capacity in cloud, the Amazon EC2/S3 propose is an infrastructure as a service product to make web-scale cloud computing much better for customers, which in this case are the developer. Recently Amazon EC2/S3 change the penalty mechanism from year penalty for monthly penalty mechanism. They guarantees 99.9% of monthly uptime percentage.

Table 4: SLA of Amazon S3

Monthly Uptime Percentage	Service Credit Percentage
Equal to or greater than 99.0% but less than 99.9%	10%
Less than 99.0%	25%

If the cloud provider delivers from 99% and less than 99.9 of the service, a 10% of the amount is reduced as a penalty, while 25% of the amount is reduced if the cloud provider delivers less than 90%.

IBM smartcloud [4] proposes a highly scalable cloud infrastructure, in addition to provide data center management approach to deal with the globe.

Table 5: SLA of IBM smartcloud

Monthly Uptime Percentage	SERVICE CREDIT
99.8%-98.8%	2%
98.8%-95.0%	5%
<95.0%	10%

The IBM smartcloud guarantees 99.8 of monthly uptime percentage. If the service provider delivers uptime percentage. If the service provider delivers less than 99.8 and more than 98.8 then the penalty will be 2% of the service credit. If the service provider deliver less than 98.8 and more than 95.0 then the penalty will be 5% of the service credit. If the service provider delivers less than 95.0 , then the penalty will be 10% of the service credit.

3. PROPOSED DYNAMIC SERVICE LEVEL AGREEMENT MODEL

In this section we presented our SLA model in 3.1, and our equation in service level agreement model in 3.2.

3.1 Static Service Level Agreement Model

Abdel-Rahman Al-Ghuwairi et al.[1] proposed a formal model that covered both sides of SLA violation penalties assessed and service level,QoS guarantees. We extended their work to provide a dynamic SLA update overtime. In this section we explain their equation.

$$SLA = \left\{ P, U, C, [t1, t2], \bigcup_{i=1}^{s} \left\{ S_i, \bigcup_{i=1}^{n} A_j \right\} \right\} \tag{1}$$

Where P is cloud provider, U is cloud user, C represent the cost of services over time[t1,t2], service type is S_i , A_j is attribute that measured. 96.50-997.99

$$A_j = \left\{ "a1", ME, \bigcup_{k=1}^{c} \{LV_k, LR_k, UV_k, UR_k, PE_k\} \right\} \tag{2}$$

Where a is attribute name such as disk storage, ME represent measurement function for a. The union represent the compound inequality. Lower bound, Lower bound Relation(LR_k), Upper bound , Upper bound Relation(UR_k). which

$$LR_k, UR_k \in \{<, >, \leq, \geq, =, \neq\}$$

.

$$ME\,(A_j) = F\,(m1, ..., mn) \tag{3}$$

Where F(m1,...,m2) represent the function defined the attributes value which computed by the function over multiple metrics. $ME\,(A_j) = \frac{V(m1)-V(m2)}{V(m1)} \times \frac{100}{100}$ where PE_k represent penalty function,VPE_k is the valuation function of user.d_k is constant value we used it where CRT_k,CRV_k =0. CRV_k is the credit value. CRT_k is the credit time factors. MCR_kMCRk is the maximum value of credit that

Table 6: Penalty function

EPVk/credit	CRTk	CRVk
dk	0	0
CRVk*C	0	0<
min(T,G*h)	0<	0<

allowed to be taken from cloud provider: where T is the maximum allowed penalty, G is the penalty for each time unit of violation, h is the number of time units of violation.

3.2 Dynamic Service Level Agreement Model (DSLA)

The Architecture of our model is Service Oriented Architecture (SOA), is easy to be understood and applied as well. Most of proposed models are not flexible enough to be updated by QoS parameter values. Updating SLA gives benefits for both cloud user and provider, users will pay less and provider will avoid more violations (which is also costly) according to the obtained values. The model has two parts for updating the SLA, the first one is to change terms according to user or provider request (mainly the provider), the second will do changes on SLA automatically according to the number of violations happened. As shown in Figure 1. we may observe that the model is composed of three main services: Monitor, analyzer and terms generator. The SLA life cycle will start with the initial SLA that will be yielded after negotiation between user and provider. After that the monitor will keep monitoring the QoS parameters like availability and sending the values for the analyzer to check if there is any violation with values compared with values agreed on initial SLA, if any violation happened on one of QoS parameter, it will be sent for user and provider and stored on database with its values. When ever a violation occurred the violation will be classified according to their values high, medium and low violations. When it is time for update as agreed on the initial SLA or the agreement get expired and need to renewal, the new terms generator propose a new values for each QoS parameter according to the stored data. For example, if the availability was 98% and it has more than 10 violations then the generator will reduce the availability to 95% to avoid more violations with a cost lower than the previous one. The new SLA proposed will be reviewed by cloud user and provider to get approved. If both agreed, then new SLA will be issued, if one of them does not approve, then the SLA will be negotiated again to get a new SLA with needed modifications.

3.3 Proposed DSLA model

In this section we proposed our equation for SLA dynamic model.

$$EP = \{VP_k, EP_k, \alpha, \beta, \sigma\} \tag{4}$$

Where EP_k is expected penalty can be written a in equation 7. where the VPE_k is violation penalty function of the kth SLA ,alpha is the average of PE_k for period of time for the user we want to update this SLA, which represent below in equation 5.

$$\alpha = \frac{\sum_{i=1}^{i=n} PE_k}{n} \tag{5}$$

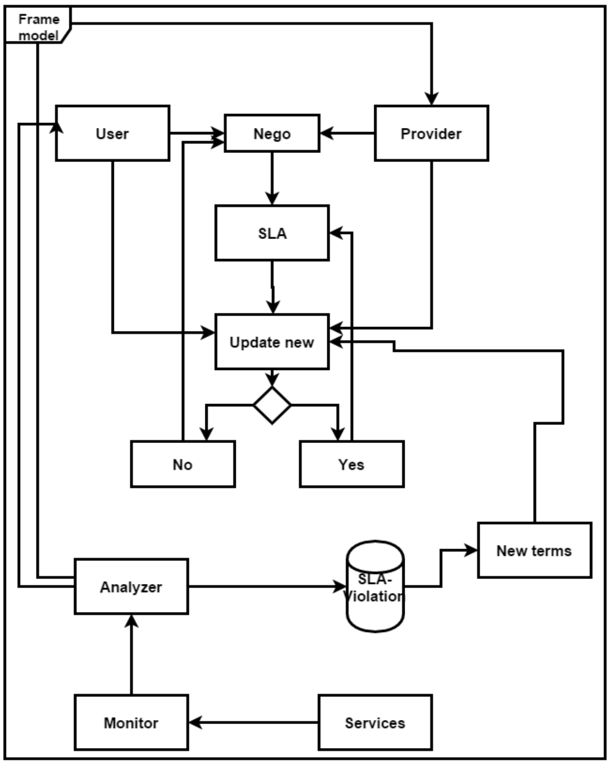

Figure 1: The dynamic SLA model

Beta is the average of PE_k for period of time for all users in the system except the user who wants to update his service level agreement, which represent below in equation 6.

$$\beta = \frac{where, \alpha' \neq \alpha \sum_{j=1}^{j=r} \alpha'}{r-1} \tag{6}$$

Sigma is the percentage which cloud provider will decrease from his benefit. To convince the cloud user to decrease the QoS guarantee in the SLA for specific attribute.The estimated penalty equation is define in equation 7

$$EP_k = \frac{\alpha + \beta}{2} \tag{7}$$

EP_k is expected penalty where it is obtained by applied the equation from 4 to equation 7, and PE_k is the value obtained by applied equation 1,2,3 and table 4.Then the proposed SLA-priced function are calculated in equation 8

$$PP_k = C - (EP_k \times \sigma) \tag{8}$$

Where the PP_k is proposed SLA-priced function of the kth SLA. C is the total billing for one month. EP_k is estimated penalty function and sigma is the percentage which cloud provider will decrease from his benefit.The new QoS guarantee percentage(NQG) represent below in equation 9.

$$NQG = min\left(\frac{j \neq k \sum_{j=1}^{j=n} \left[\frac{\sum_{i=1}^{i=n} ME_i}{n}\right]}{r-1}, ME_1, ..., ME_n\right) - .01 \tag{9}$$

Equation 9 cover the user side of tradeoff. Cloud provider will offer the new term for cloud user by reducing the cost of service and decreasing QoS guarantee.

4. EXAMPLE OF PROPOSE MODEL

In this section we describe update SLA model equation on example defined below. Assume cloudy as cloud provider company, cloudy provide IaaS (e.g.,a VM) for 20$ a month. Cloudy guarantee service availability as shown in table 4. Cloudy company will credit for the cloud user with 3 if service availability is between 99.5% and 99.9%. This provider also will credit 5% from full month amount of service billing for each 30 minutes downtime. If service availability is less than 99.5%. Assume "Tom" as a user who wants to assign an agreement with cloudy company to have a VM services, for the month March in the year 2015. To apply this example for the proposed model in this paper we will start from equation 1 provided above :

Table 7: value of the example

Monthly Uptime Percentage	Amount of Credit
99.5<=U<99.9	$3
0<=U<99.5	5% of the cost

$$SLA = \left\{ P, U, C, [t1, t2], \bigcup_{i=1}^{s} \left\{ S_i, \bigcup_{i=1}^{n} A_j \right\} \right\}$$

Then the SLA parameters can substitute as following:
P="cloudy",U="Tom", C=$20,and the billing cycle of SLA is [t1,t2]=[2/1/2015,2/31/215]. S=S1="cloud system", a1="service availability".second we will apply equation 2:

$$A_j = \left\{ "a1", ME, \bigcup_{k=1}^{c} \{LV_k, LR_k, UV_k, UR_k, PE_k\} \right\}$$

equation can substitute as following:
A1="service availability", ME,99.5%,<=,99.9%,<,PEk1, 0%,<=,99.5%,<,PEk2.table 5 summarized the example:

Table 8: Example table

Si	ai	LVk	UVK	CRVk	CRTk
VM	Ava.	99.5%	99.9%	3$	0
VM	Ava.	0	99.5%	5%	30m

to calculate the value of attribute by using measurement function introduced in equation 3:

$$ME(A_j) = F(m1, ..., mn)$$

system availability have two metrics to measure it "service downtime " and "service total hour". After mapping these attribute the evaluation function for a1 is

$$ME(A_j) = \frac{V(m1) - V(m2)}{V(m1)} \times \frac{100}{100}$$

As we mentioned above the period of SLA cycle is one month, then the total hour is 720 hours. which is V(m1). assume that the network was down for 4.5 hour which is V(m2). then ME=

$$\frac{720 - 4.5}{720} \times \frac{100}{100} = 99.375$$

this result is among compound inequality two that shown in table 3, due that we will apply equation 5 penalty function, here from the results we use the third part of penalty function. Then the parameter of penalty can be substituted as following: CRVk=%6 and C=$20

$$G_k = CRV_k \times C = \frac{5}{100} \times 20 = \$1.2$$

$$VIM = 1 - ME(a) = 1 - 0.99375$$

$$CTP_k = \frac{CTR_k}{TMB} = \frac{45}{43200} = 0.001$$

,now the violation number is

$$H = \frac{VIM}{CTR_k} = \frac{0.0069}{0.001} = 6.9\$$$

that means the credit will applied fee 7 time as we mentioned above this case used third parts of penalty function so

$$T_k = MCR_k \times C = \frac{100}{100} \times 20 = 20$$

now we will calculate the amount of penalty from equation 5

$$PE_k = min(T_k(G_k \times h_k))$$

$$PE_k = min(20(1.2 \times 7)) = 9.168$$

We applied above equation for user Tom for three months and the results was respectively 8.28, 9.168 and 7.5. after that we applied the equation above for 10 user for three months and the average for penalty for them is 8. Assume here the cloud provider will substitute sigma for 0.5, then the expectation penalty from applying equation 6 is 8.151.

$$EP_k = \frac{\alpha + \beta}{2}$$

Now, we will substitute equation eight to calculate the proposed term as following

$$PP_k = C - (EP_k \times \sigma)$$

$$PP_k = 20 - \left(8.151 \times \frac{1}{2}\right)$$

=4.75$. Cloud provider will offer the new term for cloud user by reducing the cost of service while decreasing QoS guarantee. The result of service availability attribute was respectively 99.375, 99.236 and 99.31 ,then we substitute equation 9; new QoS Guarantee term

$$NQG = min\left(\frac{j \neq k \sum_{j=1}^{j=n}\left[\frac{\sum_{i=1}^{i=n} ME_i}{n}\right]}{r-1}, ME_1, ..., ME_n\right) - .01$$

. Min(99.253,99.375,99.236,99.31)-.01=99.236-.01%=99.236% of availability. the new term will be 15.25$ the price of service for month and the availability percent= 99.236. Thus, we assume Tom well be convinced of updating his service level agreement. We here trade off between the full amount of billing which is cost and the QoS guarantees by the provider.

5. RELATED WORK

Many models have been proposed to solve SLA monitoring problems, such as SLA standard, SLA trust issues,, dynamic behaviour of QoS etc.but there are few researchers are consider a models to deal with the problem of changing SLA overtime. In this section we propose the current model of monitoring SLA. Abdel-Rahman AL-Ghuwairi et al. [6] proposed a formal model that covers both sides of service level agreement violation penalties assessed and service level agreement QoS guarantees. After describing these sides of service level agreement, the authors also describes how to enforce SLA automatically in their model. Mohammed Alhamad et al. [7] proposed a classification of the main criteria that are related to SLA in cloud comuting at design level.The authors presented the strategies of negotiation between cloud provider and cloud user, and propose a method to increase the reliability and trust between cloud provider and cloud user in negotiation stage. Knoesis et al. [9] proposed a management mechanism for service level agreement in cloud computing which used web service level agreement(WSLA) standard framework, this mechanism covers both sides of SLA monitoring and enforcement that built based on service oriented architecture. WSLA framework support by third party feature to assure the enforcement and monitoring phases and solve trust problem. This paper also validate the proposed mechanism by applying a use case at real world level. David Ameller et al. [8] proposed a monitor service level agreement violation approach, which name is SALMon.The SALMon have three main architecture components; Measurements which based on ISO 9126-1, Analyzer to verify of SLA violation rules, and decision makers which take the write action to satisfy SLA violation rules. Nirnay Ghosh et al. [11] presented an identifcation of storage as a service non-trivial SLA parameters for cloud delivery model.after identifcation process the Service level Objectives (SLOs) presented formally which predicate logic by first order. Finally they proposed a monitoring framework which is a novel framework to assure the trust, by enabling third party to achieve compliance checking. Karl Gzajkuski et al. [9] proposed a resource management level that define three type of independent resource for service level agreement. The first type delivers capability of SLA by formalizing agreements, the second type deliver activities performance, the third type define activities binding of these capabilities. The model also define an acquisition and negotiation protocol that name by SNAP which enabled reliability for resource management for remote SLAs. Finally the author explained how to develop the model in the context of globus toolkit.Farhana h. Zulkernine et al. [13] presented a novel framework of trusted negotiation broker which performs the requirement business level. The business-level requirement was mapped to decision function parameters by defining mathematical models, which decreases the complexity of using the system by the parties. Also an algorithm was defined to adapt decision function during negotiation time to response to preferences of the consumer or offers provide by an opponent's. [13] Validate their proposal by applying an extensive experimental study using prototype framework. There are a few researches that consider the cloud user side such as [6], and many other researches that consider cloud provider side. In our model we propose dynamic model that based on tradeoff between price of cloud service and the guarantee percentage of Qos parameters.

6. CONCLUSION AND FUTURE WORK

In the cloud system, the dynamic nature of QoS parameters,the desires of changing the terms from both sides cloud provider or cloud user and continuously changing of the policy of the cloud company in addition to the above mentioned reasons, there is an emerging need for changing the SLA overtime, these problems will increase rapidly in the cloud computing companies.In this paper we proposed a dynamic model for updating cloud computing SLA overtime, in order to get a flexible, adaptive SLA and to avoid costly violations. Our SLA-model covers three phases of SLA negotiation level, SLA-monitoring level and the SLA-enforcement penalty level and introduced more than one level which is SLA-level to deal with the problems that we mentioned above. We have also discussed our formal framework that define SLA-model in formal way. In the future we will add defect prediction techniques to develop the analyzer which analyzes SLA-violation database, also we will investigated about dynamic multi-cloud monitoring model.

7. ACKNOWLEDGMENTS

Our thanks to IPAC 2016 conference team for their support and help our software engineering research groups at Hashemite University in Jordan.

8. REFERENCES

[1] Sla for google app engine. www.google.com/apps/intl/en-GB/terms/sla/. Accessed: 2015-12-15.

[2] Sla in amazon s3. aws.amazon.com/ec2/sla/,2013. Accessed: 2015-12-15.

[3] Sla of dell dedicated. www.scalematrix.com/dell/sla. Accessed: 2015-12-15.

[4] Sla of ibm smart cloud. www-03.ibm.com/.../sla/. Accessed: 2015-12-15.

[5] Sla of microsoft azure. azure.microsoft.com/en-us/support/legal/sla,2015. Accessed: 2015-12-15.

[6] A.-R. Al-Ghuwairi and J. Cook. Modeling and enforcement of cloud computing service level agreements. Technical report, Technical Report, 2012.

[7] M. Alhamad, T. Dillon, and E. Chang. Conceptual sla framework for cloud computing. In *Digital Ecosystems and Technologies (DEST), 2010 4th IEEE International Conference on*, pages 606–610. IEEE, 2010.

[8] D. Ameller and X. Franch. Service level agreement monitor (salmon). In *Composition-Based Software Systems, 2008. ICCBSS 2008. Seventh International Conference on*, pages 224–227. IEEE, 2008.

[9] K. Czajkowski, I. Foster, C. Kesselman, V. Sander, and S. Tuecke. Snap: A protocol for negotiating service level agreements and coordinating resource management in distributed systems. In *Job scheduling strategies for parallel processing*, pages 153–183. Springer, 2002.

[10] A. Gandhi, Y. Chen, D. Gmach, M. Arlitt, and M. Marwah. Hybrid resource provisioning for minimizing data center sla violations and power consumption. *Sustainable Computing: Informatics and Systems*, 2(2):91–104, 2012.

[11] N. Ghosh and S. K. Ghosh. An approach to identify and monitor sla parameters for storage-as-a-service cloud delivery model. In *Globecom Workshops (GC Wkshps), 2012 IEEE*, pages 724–729. IEEE, 2012.

[12] P. Mell and T. Grance. The nist definition of cloud computing. 2011.

[13] F. H. Zulkernine and P. Martin. An adaptive and intelligent sla negotiation system for web services. *Services Computing, IEEE Transactions on*, 4(1):31–43, 2011.