2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing

An Autonomous Reliability-aware Negotiation Strategy for Cloud Computing Environments

Amir Vahid Dastjerdi and Rajkumar Buyya
Cloud Computing and **D**istributed **S**ystems (CLOUDS) Laboratory,
Department of Computing and Information Systems, The University of Melbourne, Parkville, VIC 3010, Australia,
[amirv, raj]@csse.unimelb.edu.au

***Abstract*—Cloud computing paradigm allows subscription-based access to computing and storages services over the Internet. Since with advances of Cloud technology, operations such as discovery, scaling, and monitoring are accomplished automatically, negotiation between Cloud service requesters and providers can be a bottleneck if it is carried out by humans. Therefore, our objective is to offer a state-of-the-art solution to automate the negotiation process in Cloud environments. In previous works in the SLA negotiation area, requesters trust whatever QoS criteria values providers offer in the process of negotiation. However, the proposed negotiation strategy for requesters in this work is capable of assessing reliability of offers received from Cloud providers. In addition, our proposed negotiation strategy for Cloud providers considers utilization of resources when it generates new offers during negotiation and concedes more on the price of less utilized resources. The experimental results show that our strategy helps Cloud providers to increase their profits when they are participating in parallel negotiation with multiple requesters.**

Keywords- Cloud computing; SLA negotiation;

I. INTRODUCTION

Cloud computing has transferred the delivery of IT services to a new level that brings the comfort of traditional utilities such as water and electricity to its users. The advantages of Cloud computing platforms, such as cost effectiveness, scalability and ease of management, encourage more and more companies and service providers to adopt the Cloud computing platform and offer their solutions via Cloud computing models. According to a recent survey of IT decision makers of large companies, 68 of the respondents expect that in three years, more than 50 of their company IT services will be migrated to Cloud platforms [13].

Service deployment in Cloud can be considered as a process consisting phases as shown in Figure 1. During the Service *Discovery* phase, user requirements are used as an input for discovering the best suited Cloud services among various repositories of Cloud providers. In the *SLA Negotiation* phase, discovered providers and the user negotiate on the quality of services. Next, an SLA contract will be achieved if two parties reach an agreement on a set of QoS values. Then, the acquired service will be continuously monitored in the *Monitoring* phase. If the monitoring service detects that predefined thresholds are reached, services are scaled dynamically in the *Scaling Phase*. Finally, in the

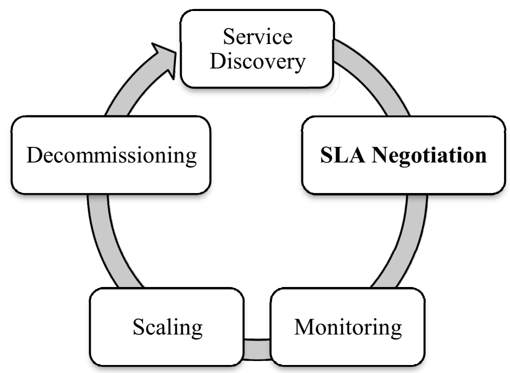


Figure 1. Service deployment phases in Cloud environments.

Decommissioning phase, last minute operations are carried out before the service is terminated. With advance of Cloud technology, operations such as discovery, scaling, monitoring, and decommissioning are accomplished automatically. Therefore, negotiations between Cloud services requesters and providers can be a bottleneck if they are carried out by humans. Consequently, this work's objective is to offer a state-of-the-art solution to automate the negotiation process in Cloud computing environments.

Cloud SLA Negotiation is a process of joint decision-making between Cloud users and providers to resolve their conflicting objectives. Cloud services have cost, availability, and other Quality of Services (QoS) on one hand, and generate profit on the other hand. In Cloud environment, both service requesters and providers have cost-benefit models for negotiation and decision-making. Therefore, SLA negotiation automation requires the mapping of knowledge and objectives of policy makers to lower level decision making techniques. The first step towards the automation is finding, capturing, and modeling goals and objectives of parties involved in negotiation and the second step is finding a proper strategy to use those goals in the low-level negotiation process.

In this work, the negotiation target is a Cloud virtual machine service and negotiated parameters are listed below.

- Hard Disk (functional requirement, and fix)
- CPU (functional requirement and fix)
- RAM (functional requirement and fix)

978-0-7695-4691-9/12 $26.00 © 2012 IEEE
DOI 10.1109/CCGrid.2012.101

- Cost (QoS requirement and negotiable)
- Availability (QoS requirement and negotiable)
- Deadline (non-functional requirement and fix)

As described in Table I, users wish lowest price and highest availability while Cloud providers would like to sell their services in the highest possible price and at the lowest QoS guarantee. Users have time constraints when they are participating in the negotiation. The reason is if they do not acquire the required resources by a particular time they are not able to satisfy their end users expectations or reach their business objectives. Furthermore, Cloud providers have to consider utilization of resources when they are offering prices during negotiation. It means that they are willing to concede on the prices of resources which are less utilized. When service requestors are conceding in such multi-issue negotiation, the negotiation strategy has to prioritize the criteria which are more important to the users. Moreover, negotiation strategies for both parties have the objectives of maximizing the chance of signing the contract during the negotiation.

TABLE I. NEGOTIATION OBJECTIVES

Objectives	Negotiation Parties	
	Requestors	***Providers***
Cost	To be minimized	To be maximized
Availability	To be maximized	To be minimized
Other	-Acquiring the resource by deadline -Conceding on less important QoS - Verifying offers reliability	-Maximizing number of agreements -Maximizing profit

Automated SLA negotiation has been studied in the context of Service Oriented Architecture (SOA) and Grid Computing. They [2, 3, 5] mainly focused on offering negotiation strategies which maximize user's utility values and number of signed contracts. However, previous works have not considered Cloud management issues (such as resource utilization balancing) in the bargaining strategy. It means that Cloud providers are willing to concede on the price of resources which are less utilized, and that has to be reflected in negotiation tactics. In addition, previous works have not considered reliability in the negotiation process. That is because service requestors would trust whatever QoS criteria values providers offer in the process of negotiation. Nevertheless, providers may offer a QoS values in negotiation which was not fully achieved according to the monitored QoS data. Consequently, this work presents a negotiation strategy that captures user's preferences and provider's resource utilization status, and utilizes a time-dependent tactic along with theory of statistics to maximize Cloud providers' profit, adhere to deadline constraints of users and verifies providers offer reliability.

The rest of the paper is organized as follows: Section two highlights challenges in SLA negotiation in Cloud that have not been addressed by previous works. Then, while the third Section provides a high-level description of the negotiation framework for the readers, the forth Section covers the detail of negotiation strategies for both Cloud providers and users. In Section five the negotiation strategies are tested to prove their effectives, and applicability. Next, Section six aims at highlighting the uniqueness of the proposed negotiation framework by comparing its characteristics with related work. Finally, the work is concluded with the suggestions on future works.

II. CHALLENGES

A. Offers reliability

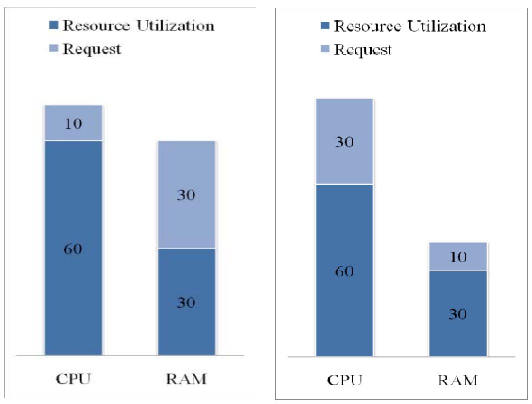
In negotiation models offered in frameworks such as [2, 3, 5], a method that can determine the reliability of offers and counter offers is missing. Since in the parallel negotiation a party makes decision based on the presented QoS values in SLA offers, there has to be a way to know how reliable the provider is in delivering those promised QoS values. The recorded data from monitoring services can be analyzed and converted to reliability information of offers. Monitoring data determines to which extent an SLA is achieved and facilitates a procedure taken by a user to receive compensation when the SLA is violated. The monitoring is based on the copy of signed SLAs which is kept in a SLA repository. Third-party monitoring results can be similar to what Cloud harmony [14] services report. To make inference from the observed data we use the theory of statistics (Beta Density Function) which will be explained in Section IV.

B. Balancing resource utilization

When service providers are concurrently negotiating with multiple users, the majority of previous works [5] assume that using the same negotiation strategy for all incoming requests maximizes providers' profits. However, we argue that in the Cloud context, providers are interested in a SLA negotiation strategy that balances the available resources, which helps them to host more virtual machines. To achieve that providers have to concede more on the price of the resources that are less utilized (or have more free capacity) and less on the price of resources which are more utilized. Consequently, providers offer more attractive prices in earlier stages of negotiation for clients whose requested virtual machines' allocation would balance resource utilization. For example, the request shown in Figure 2(a) will be offered more attractive price in early stage of negotiation compared to the request shown in Figure 2(b).

III. NEGOTIATION FRAMEWORK

Figure 3 shows major components in the negotiation framework and Figure 4 briefly describes the sequence of interaction between the Cloud and the requester negotiation services. A service requester specifies certain requirements such as hardware specifications like CPU, storage, and memory. In addition, the requester provides preferences on the QoS criteria. Then, functional and QoS requirements are used as input for discovery of suited Cloud services. Afterwards, client Negotiation Service (NS) starts the negotiation with discovered service providers' NSes on QoS criteria (price and availability) based on the requester's preferences. It is worth mentioning that the client's budget and deadline for acquiring resources are used by the client NS to make a decision on accepting or rejecting an offer. Client NS uses a time-dependent tactic which takes client's preferences as an input and automatically generates an initial and then consequent offers. Once the Cloud NS receives the offer, it uses request functional, QoS requirements, and Cloud resources utilization supplied by monitoring system to generate counter offers. On the arrival of providers' offers, the client NS uses reliability evaluator components and a time-dependent tactic to accept or reject the offer, or otherwise reply with a counteroffer.



(a) (b)

Figure 2. Requests and their effects on balancing resource utilization.

If the negotiation is successful, an SLA contract is signed by both parties and the obtained contract, which includes set of expected QoS values (service level objectives), is kept in the SLA contract repository. The monitoring service manager discovers and selects the required monitoring services based on user preferences (cost and reliability) for each SLA contract in the repository. They constantly monitor the SLA and notify reliability evaluator on any violation of service level objectives.

Moreover, as explained in our previous work [11], QoS ontology, Cloud services, their QoS, user requests, and SLA contracts are described using Web Service Modeling Language (WSML) [12]. Consequently, Cloud service and monitoring service discovery components can perform semantic matching using terms defined in the ontology. The ontology-based discovery can increase the level of flexibility and automation, allowing the two parties to use their own terminology as long as it is related to the commonly understood conceptual model.

IV. NEGOTIATION STRATEGY

Prior to explain the negotiation strategies for each party, a brief description of the negotiation model and the applied negotiation tactics are given. In addition, descriptions of symbols used for express negotiation process are listed in Table II.

A. Negotiation model

To create a negotiation model, we extended the model proposed in [1] to incorporate the reliability of offers. In the model, the negotiation service receives requestor preferences on the importance (W_i) of n negotiation issues, min_i and max_i (reservation values) which are acceptable range of values for issue i (VI_i), and negotiation deadline (t_{max}). The service then measures the utility of offers received from other negotiation service based on the Equation 1.

$$UV = \sum_{i=1}^{n} W_i VI_i \text{ Where } \sum_{i=1}^{n} W_i = 1 \quad (1)$$

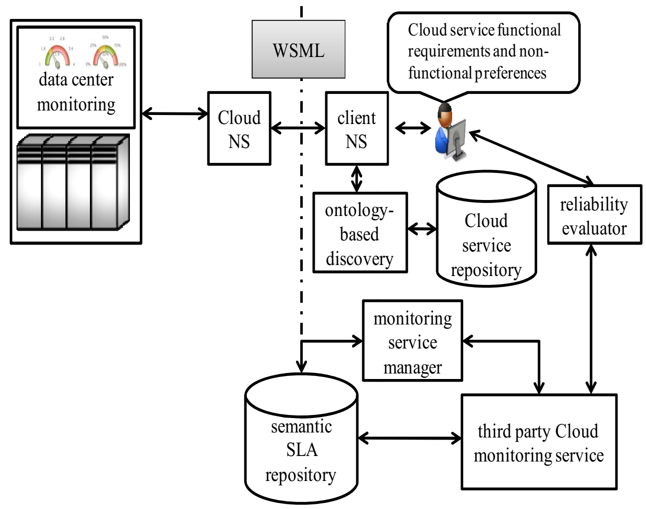


Figure 3. Service negotiation framework.

Next, as shown in Equation 2, the offer will be only accepted if its value is greater than or equal to the utility of the counter offer that will be sent by the negotiation service. Otherwise the negotiation service generates a new counter offer. In addition, if the timestamp of the received offer (t_{offer}) is greater than the deadline the service terminates the negotiation.

$$Response = \begin{cases} terminate & if \quad t_{offer} > t_{max} \\ accept & if \quad UV_{offer} > UV_{counter\ offer} \\ generate\ new\ counter\ offer & otherwise \end{cases} \quad (2)$$

A. Time-dependent negotiation tactic

As cited by Faratin et al. [2], time-dependent negotiation tactics are a class of functions that work out the value of an issue by considering the time factor. Therefore, they are particularly helpful when NS receives a deadline (t_{max}) as an input, and has to concede faster as the deadline approaches. For this family of tactics, Equation 3 is used by NS "*a*" which represents either a Cloud service requestor or a provider to generate a new counter offer for NS "*b*" for negotiable issue *i*.

$$O_{a\to b}^{t}[i] = \begin{cases} min_i^a + \alpha_i^a(t)(max_i^a - min_i^a) \\ \quad if\ V_i^a\ is\ decreasing \\ min_i^a + (1-\alpha_i^a(t))(max_i^a - min_i^a) \\ \quad if\ V_i^a\ is\ increasing \end{cases} \quad (3)$$

TABLE II. DESCRIPTION OF SYMBOLS

symbols	Description
a, b	negotiation parties
W_i	importance of issue *i*
VI_i	offer value for the issue *i*
UV	utility value of the offer
t_{offer}	offer time stamp
t_{max}	negotiation deadline
$O_{a\to b}^{t}[i]$	offer sent from *a* to *b* for issue *i*
min_i^a	minimum acceptable value of issue i for *a*
max_i^a	maximum acceptable value of issue i for *a*
$\alpha_i^a(t)$	time-dependent function of issue *i* for *a*
V_i^a	value offered for issue *i* by *a*
k_j^a	initial offer value for the issue *i* by *a*
β	convexity degree
P_t	price of virtual machine instance at *t*
RP_{jt}	price of a resource *j* (e.g. RAM) at *t*
αRp_j	time dependent function for price of resource *j*
IRP_j	initial price for resource *j*
A_j	portion of resource *j* that is available
β_j	convexity degree for price of resource *j*
$RUBO$	resource utilization balancing oriented tactic
PO	priority oriented tactic
γ_1	relative importance of *RUBO*
γ_2	relative importance of *PO*
$RC_{offerVI_i}$	reliability constraint for issue *i*
$R_{offerVI_i}$	reliability of an offer's value of issue *i*

Numerous functions have been defined for calculation of $\alpha_i^a(t)$ such as polynomial and exponential [2]. As it can be figured out from Equation 4, by changing the value of β(convexity degree) in both functions the behavior of the negotiation tactic changes. If $\beta > 1$ then the tactic reaches its reservation's value at the early stage of negotiation. However, in the case of $\beta < 1$ it concedes to its reservation value only when the deadline is approaching. In our work we adopt this family of the negotiation functions and change β dynamically to maximize the NS utility function.

$$\alpha_i^a(t) = \begin{cases} k_i^a + (1-k_i^a)\left(\frac{min(t,t_{max})}{t_{max}}\right)^{1/\beta} & Polynomial \\ e^{(1-\frac{min(t,t_{max})}{t_{max}})^{\beta}(lnk_i^a)} & Exponential \end{cases} \quad (4)$$

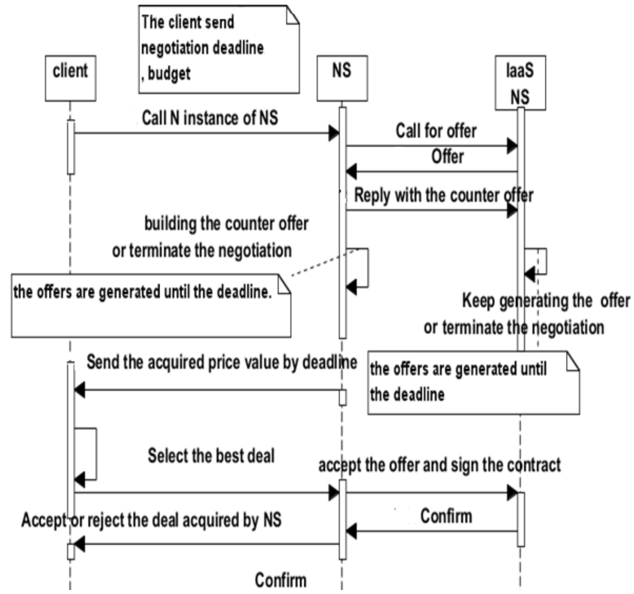


Figure 4. Negotiation sequence diagram.

B. Providers Strategy

For providers, the negotiation service input is composed of the Cloud resource utilization, minimum and maximum resource prices, and amounts of requested resources. The output of NS can be a SLA contract with full specification of provider, client, services, and service level objectives. Providers are interested in an SLA negotiation strategy that balances the available resources, and gives the attractive offers while keeps their utility functions high. To achieve that we have to concede more (by adjusting time-dependent function parameters) on the price of the resources which are less utilized (or have more free capacity) and less on the resources which are more utilized.

There are works that requires time-dependent function parameters to be given explicitly. However, Zulkernine et al. [5] proposed a method to derive the parameters from the high level negotiation policy. Inspired by their work, we propose an approach to derive a price for the next offer based on the Cloud resource utilization. In comparison to Zulkernine et al. work, we argue that our approach is suitable for parallel negotiation in Cloud context. The reason is we are discriminating regarding the pattern of concession when negotiating concurrently with multiple clients, while Zulkernine et al. applies same pattern of concession for all clients. As shown in Equation (5-9) we first define a total price of an instance as the sum of prices of its individual resources (Equation 5). In the next step, for each resource, a time-dependent function (Equation 6 and 7) is defined, and its parameter is adjusted (Equation 8 and 9) based on that particular type of underutilized resource capacity compared to average resources' unoccupied capacity ($\bar{A}$).

$$P_t = \sum_{j=1}^{m} RP_{jt} \quad (5)$$

$$RP_{jt} = MinRP_j + \alpha RP_j(MaxRP_j - MinRP_j) \quad (6)$$

$$\alpha Rp_j = IRP_j(1 - IRP_j) + \left(\frac{min(t,t_{max})}{t_{max}}\right)^{1/\beta_j} \quad (7)$$

$$\bar{A} = \frac{\sum_{j=1}^{m} A_j}{k} \quad (8)$$

$$\beta_j = e^{C(A_j - \overline{A})} \quad (9)$$

As shown in Equation 9, when the free capacity of a resource is higher compared to the average resources' free capacity in the Cloud, $A_j - \bar{A} > 0$ and β_j>1, and therefore the negotiation strategy is conceding on the price of that resource. As a result, providers offer more attractive price in earlier stages of negotiation for clients whose requested virtual machines' allocations would balance resource utilization. This increases the chance of reaching the agreement with the preferred request. However, in this tactic β is calculated based on the resources utilization and does not reflect the preferences of provider regarding the importance of price and guaranteed availability criteria. The tactic offered in [5] is used in Equation 10 to derive β from provider's preferences. Consequently, in order to satisfy all providers' objectives, the negotiation strategy has to be built as a mixture of those aforementioned tactics as shown in Equation 11.

$$\beta_i = e^{c(\frac{1}{n} - W_i)} \quad (10)$$

Where n is the number of criteria in the negotiation, c is a constant, and W_i is the importance of issue *i* and $\sum_{i=1}^{n} W_i$=1.

$$O_{a\rightarrow b}^{t}[i] = \gamma_1 RUBO_{a\rightarrow b}^{t}[i] + \gamma_2 PO_{a\rightarrow b}^{t}[i] \quad (11)$$

Where $\gamma_1 + \gamma_2 = 1$, and RUBO, PO are offers' issue values generated by Resource Utilization Balancing Oriented tactic and Preference Oriented tactic respectively.

C. Cloud Client NaaS

The client NS receives user references on budget, deadline, and QoS criteria importance. Then it maps the user preferences to low level time-dependent parameters as described in the previous section based on the Equation 10 [5]. It means that β is defined in a way that the NS concedes less if the criteria are more important to the user and concedes more otherwise. Analytic Hierarchy Process (AHP) [11] is adopted in our work to capture the importance of the criteria for the user. Similar to the provider NS, the output can be a SLA contract with full specification of services, provider, client and service level objectives. In this strategy our contribution lies in probabilistic assessment of offers reliability in negotiation.

Client NS assesses providers offers' in a probabilistic approach based on their past adherence level to the SLA contracts. Therefore, as shown in Equation 12, the client NS only accepts offers when similar previous accepted offers have achieved certain level of reliability (based on the monitored data) for each issue. For example if in multi-issue negotiation, a provider concedes in availability and its reliability in availability is not high users should not consider that as an attractive offer.

$$offer\ acceptance\ conidtions = \begin{cases} 1)\ if\ \ UV_{offer} > UV_{counter\ offer} \qquad and \\ 2) if\ for\ each\ VI_i \quad R_{offerVI_i} > RC_{offerVI_i} \end{cases} \quad (12)$$

We utilized the beta reputation system [4] to assess the reliability of offers. The reason is that Monitoring Outcome (MO) of a particular SLA contract can be modeled as Equation 13, and therefore is a binary event. Consequently, the beta density function, which is shown in Equation 14, can be efficiently used to calculate posteriori probabilities of the event. As a result, the mean or expected value of the distribution can be represented by Equation 15.

$$MO = \{SLA\ not\ violated, SLA\ violated\} \quad (13)$$

$$f(x \mid \rho, \tau) = \frac{\Gamma(\rho+\tau)}{\Gamma(\rho)\Gamma(\tau)} x^{\rho-1}(1-x)^{\tau-1} \quad (14)$$

where $0 \le x \le 1,\ \rho < 0,\ \tau > 0$

$$\mu = E(x) = \rho/(\rho + \tau) \quad (15)$$

As mentioned earlier in the Section III, in our architecture a component is responsible for monitoring SLA contracts. If we assume that the monitoring component has detected that SLA violation occurred *v* times for provider of *p* (for total number of *n* monitored SLAs). Consequently, considering that $\rho = n - v + 1$ and $\tau = v + 1$, the reliability is equal to probability expectation that SLA is not going to be violated and is calculated as shown in Equation 16. Once $R_{offerVI_i}$ is calculated for all issues, NS can only accept the offer if for all the issues $R_{offerVI_i}$ is greater than $RC_{offerVI_i}$.

$$R_{offerVI_i} = \frac{n-v+1}{n+2} \quad (16)$$

V. Performance Evaluation

We have extended CloudSim [15] with a negotiation service for client brokers and data centers. The extended negotiation package enables datacenters and negotiation broker to have distinct negotiation strategies. The designed negotiation scenario for the experiments consists of 4 client brokers and 2 data centers which offer virtual machine instances similar to Amazon EC2 offers [16]. Negotiation brokers simultaneously send requests to data centers. A request for a resource is formally represented by Equation 17.

$$\text{Request} = N_{CU} + N_{RU} + N_{HDU} \quad (17)$$

Where N_{CU} is number of CPU units requested, N_{RU} is number of RAM units requested, and N_{HDU} is number of hard disk units requested. Requests for resources generated for experiments can be classified into two classes, namely balanced and unbalanced. In a balanced request, N_{CU}=N_{RU} =N_{HDU}, while in an unbalanced requests $N_{CU} \neq N_{RU} \neq N_{HDU}$. The balanced and unbalanced requests for our experiments have been designed according to Amazon EC2 instances types [16]. Random requests (with different budgets, reliability constraints, and requested virtual machine Type) are simultaneously sent by client brokers in intervals of 60 seconds. In addition, both client NS and Cloud NS use polynomial functions.

A. Experimental Results

In this subsection, performance we measure performance (regarding profit) of proposed strategies for workloads with different number of balanced and unbalanced requests is presented. For these experiments requests' deadlines are fixed. Moreover, it is assumed that Cloud providers have similar initial prices for their resources in offers. The aim of this experiment is to investigate how successful the proposed strategy for Cloud NS is in accommodating more requests and thus increasing Cloud providers' profits which is calculated based on the number of the VM allocated and achieved price in the negotiation. The experiment in this section was repeated 30 times. In addition, profits are normalized to be in range of 0 to 100.

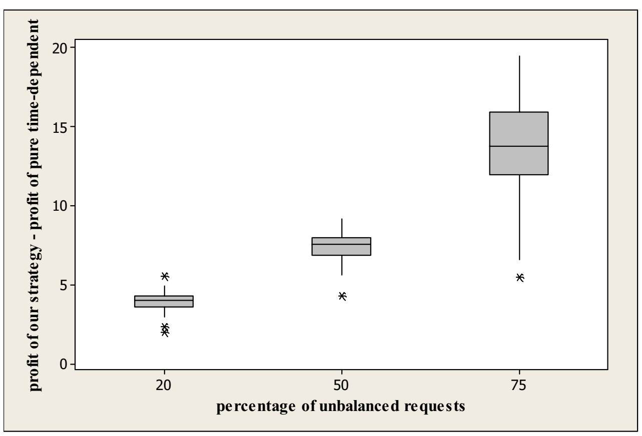


Figure 5. Comparison of resource utilization balancing oriented and purely time-dependent strategies.

In the experiment, one of data centers adopted purely time-dependent function and the other adopted our strategy. As Figure 5 shows, when percentage of unbalanced requests increases, the difference between strategies offered in works such as [5, 2] (purely time-dependent) and our work grows. The results show that even for the cases where only a small percentage of incoming requests are unbalanced (20 percent), still data centers can increase their profits by almost 4 percent on average. In addition, if the chance of a request to be unbalanced is 50 percent, then the profit growth increases to 7.5 percent on average. And finally, for the time that percentage of unbalanced requests is set to 75, our strategy can dominate previous work's strategy by nearly 14 percent.

VI. Related Work

Service Level Agreement Management (SLAM) has been investigated heavily by researchers in the context of Grid computing and Service Oriented Architecture (SOA). As shown in Figure 6, researches in this area mainly have focused on three aspects of SLAM namely as: SLA language specification [11], SLA negotiation techniques [5], and SLA monitoring approaches [10, 11].

Kotsokalis et al. [9] offer a generic architecture, which can be utilized across different domains and use cases to support SLA management and tries to address all phases of the SLA management lifecycle, from negotiation and establishment to termination, with respect to the existence of SLA interdependencies. The goal is to ensure that operations of existing services would not be affected. In contrast with our approach, no mechanism is provided for semantic annotation of SLA contract, and no details are provided on how negotiation has to be carried out when there is time and resource constraints. In addition, in their architecture there is no component to measure reliability of offers to enhance the quality of decision making in the negotiation.

In SOA context, an approach to provide support for reaching agreements between the service consumer and providers is crucial. However, reaching the agreement is more challenging in the case of service composition where the consumer negotiates with multiple providers. Yun et al. [6] has tackled this issue by adopting agent technology and extending the Foundation for Intelligent Physical Agents (FIPA) protocol [7]. The work chooses the utility function for developing decision making model for agents and uses the concept of fuzzy similarity [3] in its negotiation strategy. The presented negotiation strategy makes effort to increase the chance of reaching the agreement by generating more similarity between offers and counter offers. The language used for negotiation is WSDL, and the prototype implementation shows the feasibility of the approach. Compare to their work, our work uses WSMO and WSML to model an SLA language which brings a common understanding on QoS criteria for all parties involved in negotiation. In addition, the dose not define how the negotiation strategy of service provider differs from client when both resource and time constraints exist in the negotiation problem.

When SLA negotiation consists of multiple issues such as service time and price, approaches which are purely time-dependent [2] are no longer effective. Coehoorn et al. [8] offered a multi-issue negotiation approach by gathering information regarding opponents' preferences across negotiation issues using kernel density information. After obtaining the preferences, the work uses fuzzy similarity [3] to create a counter offer which keeps the high utility for the negotiation agents and improves the utility of the opponent. Fuzzy similarity technique was proposed by Faratin et al. in 2002 [3]. It investigates the situation where an agent is willing to offer more attractive SLA contract to its opponent, however it does not intend to decrease its utility function. The work presented an approach that determines how much (in a scale of 0 to 1) two contracts match. Having that and

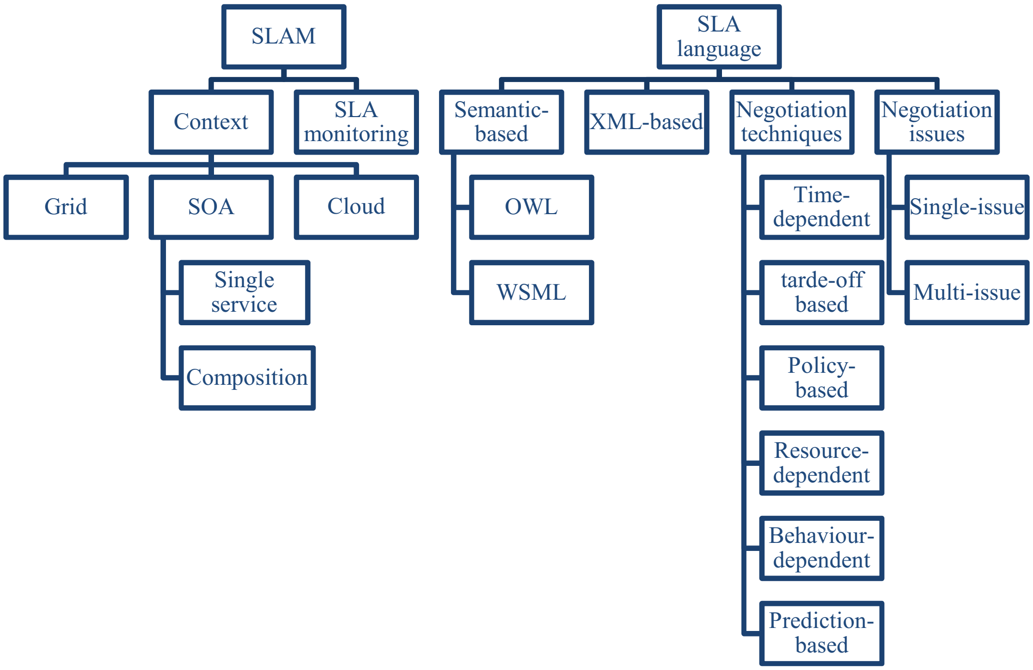


Figure 6. SLA Negotiation Taxonomy

using hill-climbing algorithm, the work searches the space of feasible contracts to find a contract that has the highest similarity with the opponent offer and still has the same utility value as the previously offered contract. Investigating effectiveness of applying fuzzy similarities in our problem is considered for our future work.

Compare to the work done by Zulkernine et al. [5], our work used WSMO and WSML to model the SLA language, which brings a common understanding on QoS criteria for all parties involved in the negotiation. In addition, we argue that our approach is suitable for parallel negotiation in Cloud context. The reason is that we are discriminating regarding the pattern of concession when negotiating concurrently with multiple clients (to increase Cloud providers profit) while Zulkernine et al. believe in having the same pattern of concession for all clients. When the parties in negotiation have deadline on trading the services, the remaining time is the most significant factor to determine the offers in each round of negotiation. In the context of Cloud computing, there are factors which have to be considered to generate offers. The reason is that Cloud service providers have to decide on the negotiation issues based on the availability of resources and time passed in negotiation.

Identifying SLA QoS parameters and fine-grain metrics to measure them is a fundamental task in the process of SLA management. Lawrence et al. [18] proposed an approach for SLA management which is built using WS-agreement and is a part of project called OPTIMIS. Their work is designed for environments where SLM is needed between service providers and infrastructure providers. In comparison to our work, they have defined several novel negotiation criteria, but not any negotiation tactics that can be utilized in Cloud environments. Similarly, Goiri et al. [17] has worked out a fine-grain QoS metric for CPU performance that can be used in SLA, and avoids fake SLAs violation to help providers achieve higher resource utilization. The major drawback of current works is that the service requestors would trust whatever QoS values providers offer in the process of negotiation, and then they signed the contract based on those values. However, providers may offer a QoS values in negotiation which is less likely to be achieved based on the monitored QoS data. The SLA negotiation works main weakness is that they have not differentiated the decision making phase from the bargaining phase. Even though they have succeeded in offering effective bargaining approaches, however they blindly trusted offers from the service providers and most importantly they make the decisions based on those offers.

VII. Conclusions and Future Directions

In this paper, we described SLA negotiation challenges in Cloud computing environments. We also proposed a time-dependent negotiation model to tackle such challenges. We have included reliability assessment of offers in the negotiation process to increase the dependability of our strategy while filling the gap between decision making and bargaining. Even though many of the works in literature apply the same pattern of concession for all clients when negotiating in parallel, we argued that discriminating regarding the pattern of concession helps Cloud providers to accommodate more requests and thus increase their profit. Our approach was tested against purely time-dependent approaches, and it shows its dominance in generating more profit for providers.

We have run other experiments by modifying parameters such as deadline of requests and initial offer values, however due to lack of space we could not present it in this paper and will be presented in a technical report. In addition, in future we are going to investigate effects of replacing the polynomial with exponential function on the profit of providers. Moreover, the effects of heterogeneous negotiation (not from same family of time-dependent tactics) on the profit of each party in negotiation will be explored. Finally, consequence of variation in reliability constraints on the number of successful negotiation is going to be examined.

ACKNOWLEDGMENTS

We thank Rodrigo N. Calheiros and Jason Read for their assistance and constructive suggestions to enhance the quality of this paper.

REFERENCES

[1] H. Raiffa, The Art and Science of Negotiation, Harvard University Press, Cambridge, USA, 1982.

[2] P. Faratin, Automated service negotiation between autonomous computational agents, PhD Thesis, 2000.

[3] P. Faratin, C. Sierra, and N. R. Jennings, "Using similarity criteria to make issue tradeoffs in automated negotiations. ", Artificial Intelligence, 142(2):205--237, 2002.

[4] A. Jøsang and R. Ismail, "The beta reputation system", Proceedings of the 15th Bled Electronic Commerce Conference, Bled, Slovenia, June 2002.

[5] F. H. Zulkernime and P. Martin, "An adaptive and intelligent SLAnegotiation system for Web servcies", IEEE Trans. on ServiceComputing, volume 4, number1,pages 31-43, 2011.

[6] J. Yan, R. Kowalczyk, J. Lin, M.B. Chhetri, S.K. Gohand J. Zhang, "Autonomous Service Level AgreementNegotiation for Service Composition Provision", Future Generation Computer Systems, vol. 23, no. 6, pp. 748–759, 2007.

[7] FIPA Iterated Contract Net Interaction Protocol Specification, http://www.fipa.org/specs.

[8] R. M. Coehoorn, N. R. Jennings, "Learning an Opponent's Preferencesto Make Effective Multi-Issue Negotiation Trade-Offs", Proceedings of the 6th International Conference on Electronic Commerce, ICEC 2004, Delft, The Netherlands, October 25-27, 2004.

[9] C. Kotsokalis, R. Yahyapour, MAR. Gonzalez, "SAMI: The SLA Management Instance", Proceedings of Fifth International Conference on Internet and Web Applications and Services, 2010, Barcelona, Spain.

[10] V. C. Emeakaroha, I. Brandic, M. Maurer, S. Dustdar, "Low level metrics to high level SLAs-LoM2HiS framework:Bridging the gap between monitored metrics and SLA parameters in Cloud environments", Proceedings of InternationalConference on High Performance Computing and Simulation (HPCS), IEEE, June 28 - July 2, 2010, Caen, France.

[11] A. V. Dastjerdi, S. G. H. Tabatabaei, and R. Buyya, "A Dependency-aware Ontology-based Approach for Deploying Service Level Agreement Monitoring Services in Cloud", Software: Practice and Experience, vol. 42, no 4, pp. 501–518, 2012.

[12] WSMO Working Group. Available from: http://www.wsmo.org.

[13] B. Narasimhan, R. Nichols, "State of cloud applications and platforms: The cloudadopters' view.", Computer 44(3), 24–28 , 2011.

[14] Cloudharmony. [Online]. http://Cloudharmony.com

[15] R. Calheiros, R. Ranjan, A. Beloglazov, C. De Rose, and R. Buyya, "Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms" Software: Practice and Experience, vol. 41, no. 1, pp. 23–50, 2011.

[16] Amazon Elastic Compute Cloud (Amazon EC2), . [Online]. http://aws.amazon. com/ec2

[17] I. Goiri, F. Julia, J. O. Fito, M. Macıas, and J. Guitart, "Resource-level QoS metric for CPU-based guarantees in cloud providers", Proceedings of the 7th international conference on Economics of grids, clouds, systems, and services (GECON'10), Ischia, Italy, 2010.

[18] A. Lawrence, K. Djemame, O. Wäldrich, W. Ziegler and C. Zsigri, "Using Service Level Agreements for Optimising Cloud Infrastructure Services", Proceedings ServiceWave Workshops, Ghent, Belgium, 2010.