ISSN: 0975-766X
CODEN: IJPTFI
Available Online through
Research Article
www.ijptonline.com

A TIME SLOT AND COST NEGOTIATION METHOD FOR CLOUD SERVICES

Venugopalan S[#1], Ramya[#2]
[1,2]Computer Science Engineering Department, Assistant Professor,
Indira Gandhi College of Engineering and Technology (Affiliated to Anna University),
Chengalpattu – Kancheepuram High Road,
Athur post, Kancheepuram District-603107, Tamilnadu, India.
Email: svg3612@gmail.com

Received on: 03-02-2017 *Accepted on: 12-03-2017*

Abstract

When we are going to make booking for Cloud services, then the customer and service suppliers have to create service level contracts during negotiation. Where it is significant for equally a customer and a service supplier to reach an agreement on the cost of a services while they are going to cloud services, to time, there is little or no negotiation maintain for mutually Time Slot and Cost Negotiations (*TSCNs*) for cloud service reserving. This editorial presents a many issue negotiation method to make possible the following: 1) *TSCNs* between Cloud mediators and 2) tradeoff between Time Slot and Cost effectiveness. Unlike several previous negotiation scheme in which a negotiation mediator build only one offer at a time. In our effort, mediator constructs several offers in a negotiation round that produce the similar collective utility, differing only in terms of individual Time Slot and Cost effectiveness. An additional innovation of effort is formulating a new time-slot usefulness function which differentiates preferences for dissimilar time slots.

Keywords: Automated negotiation, Cloud negotiation, Cloud resource allocation, many-issue negotiation, negotiation mediator, resource management.

I. Introduction

Cloud is a corresponding and disseminated system consisting of a collecting of interrelated and virtualized computers that are actively conditioned and presented as one or extra integrated calculating resource(s) based on service-level agreements launched through negotiation between service suppliers and consumer [1]. An *SLA* is a service assurance scheme which describes a set of quality of service (*QoS*) limitations such as expenditure or time limitations and indicates how the service is recommended [2]. To launch a contract among a consumer and service supplier for developing a Cloud service, several of the critical concerns consist of the following: 1) establishing when to utilize a service and 2) establishing the cost of the service [3]. Although these concerns are more critical, methods to mechanize the negotiation of time slot and cost for Cloud services have not been developed [4]. While earlier works have arrangement with advance reserving believing bandwidth or time constraints [5]–[6] and measured *SLA* negotiation [7], to time, there is no service reservation scheme that regard as equally Time Slot along with Cost negotiations (*TSCNs*). Since there is an inverse relationship between Time Slot and Cost effectiveness, time slot and cost have to be negotiated concurrently. This effort considers a several concern negotiation mechanism for *TSCNs* for Cloud service reserving [8]. In this paper, we present a novel time-slot mechanism which is designed to model the customers' and service suppliers' preferences for various time slots. In common, a customer can have various sets of satisfactory time-slot favorites. Time-slot effectiveness function consisting of multiple partial functions is used to model a customer's preferences for multiple sets of acceptable time slots. This work regard as bilateral negotiation s between a customer and service supplier, where both mediators are sensitive to time and adopt a time-dependent concession-making strategy for *TSCNs*. Since both mediators negotiate on both Time Slot and Cost, generating a counterproposal can be making either a concession or a tradeoff between time slot and cost. Therefore, a mediator's strategy for many issue negotiation is implemented using both the following:

1) *Tradeoff Algorithm:* The novelty of this work is adopting a novel tradeoff algorithm, called a "burst method" suggestion, which is planned to improve equally the negotiation velocity and the collective effectiveness.

2) *Concession-Making method:* The concession-making method decides the amount of concession for every negotiation around, which communicates to the decrease in a mediator's, estimated overall effectiveness.

We developed Mediator-Based Cloud Testbed approach which gives simple service discovery functionality through message passing. The mediator-based Cloud testbed is planed and implemented with the help of Java and the Java mediator development framework. In a Cloud market, there are many clients and service suppliers, and thus, the Cloud testbed has client mediators and supplier mediators acting on behalf of customers and suppliers. A customer mediator broadcasts a message indicating the name of the Cloud service that the customer needs to all supplier mediators, and a supplier mediator who has the service replies to the customer. In this paper, our contribution of works as follows:

1. We develop *TSCNs* mechanism that includes the design of a novel utility function for time-slot preferences.
2. We design tradeoff and concession algorithms for the concession strategy between clients and suppliers.
3. We implement an mediator-based Cloud testbed for evaluate the *TSCN* mechanism

The rest of the manuscript progress as pursues: In Section I, we properly begin the system method and the plan of Cost Price scheme for cloud services via letting. Here Section II, we discussed regarding related effort for accepting the earlier work. In section III, we presented planned scheme and their structure for performance of cost-price negotiations. In section IV, we show implemented result through screenshot and tables. Finally, we conclude overall work in Section V. We provide the acknowledgment in Section VI and reference in section VII.

II. Related Work

In existing systems, Many Issue SLA Negotiation approach was proposed by Czajkowski et al. [1] which are generalized resource management model in which resource interactions are mapped onto a well-described set of platform-independent *SLAs* that formalizes agreements to deliver capacity, perform activities, and bind activities to capabilities, respectively. The model is based on a Service Negotiation and Acquisition Protocol, which manages the lifetime management of *SLA*. For *SLA* specifications, a Meta negotiation was proposed by Brandic *et al.* [2] To manage Paschke *et al.* [11] planned a declarative rule based *SLA* language for relating *SLAs* in a common way. *RBSLA* is machine readable and has executable contract specifications. Whereas [3], [4], and [5] do not focal point on identifying negotiation approaches nor planning usefulness purposes for every negotiation term, Yan *et al.* [6] adopted a tradeoff algorithm for multi-issue *SLA* negotiations. Yan *et al.* [7] planned a structure for a Web service composition to facilitate gives *SLA* negotiation for *QoS* limitations. Based on this framework, a utility-function-based decision-making model that supports coordinated negotiation was presented. To ensure coordinated *SLA* negotiation, the framework in [9] consists of a coordinator mediator (*CA*) and a set of negotiation mediators. The *CA* is responsible for governing the composition of the *SLA* negotiations as an entire, while each negotiation mediator is in charge of the *SLA* negotiation for one service in the composition. The negotiation mediator presented in [8] is related to the design of *TSCN* mechanism. Similar to *TSCN* mechanism, the design of the decision-making model of negotiation mediator for the many issue negotiation in [9] includes the formulation of effectiveness function and a negotiation strategies. The difference between [10] and this work is that, while mediators in [10] generate a single proposal in each round using heuristics, *TSCN* mediators concurrently generate multiple proposals in each round. The time-slot utility functions in this paper enable client and service supplier mediators to express preferences for unlike time slots. Yan *et al.* [11] planned a concession-making algorithm and a tradeoff algorithm for generating counterproposals. The concession-making algorithm searches within an acceptance range for a point with a fixed deduction of the overall utility value of the current offer to generate a new counterproposal. Whereas the concession algorithm in this work adopted the time-dependent strategies in [12], a fixed deduction is preconfigured by the *NA* in [13]. Finally, it should be noted that several of the preliminary results of this work were presented in [14]. This work has significantly and noticeably augmented and generalized the preliminary work in [14] as follows. While [15] omitted any details on the time slot utility function, in this paper, a very detailed formulation of the time-slot function is provided. Moreover, being a little conference paper, [16] has omitted comparison with related works. This paper gives a much featured comparison between this work and related works on advance reservation, concurrent negotiation, *SLAs* negotiation, and many issue negotiation.

III. Proposed Scheme

In this section we implement our method and discuss about their feature follow as: in Section A, We introduce with cloud pricing through negotiation and show the advantages of this scheme. In section B, we developed cloud-based Testbed mediator for communication customer and service supplier. In section C, we implement Time Slot and Cost Negotiations with help for trade-off and Concession-making algorithm.

A. Cloud Pricing through Negotiation: Challenging concerns in Cloud service booking is devising a suitable pricing method. Amazon elastic Cloud computing (*EC2*) offers consumers with equally permanent costing and flexible costing (spot instances) for rental virtual machine (*VM*) occurrences. On-demand occurrences permit consumers to pay a permanent cost by the hour without a long-term obligation and to begin the occurrences immediately. Among reserved occurrences, consumers require to pay a one-time payment for a one or three year's term but profits from paying a low-priced hourly usage payment within the term. Spot occurrences enable consumers to offer for unused computing capability. Occurrences are charged at the spot cost set by Amazon. The spot cost alters periodically, depending on the supplier and demand for spot occurrences. Consumers' requests can be satisfied if their highest offer expenses are over the spot cost and they can run their request on the spot occurrences for as long as their highest bid costs exceed the present spot cost. The profit of spot occurrences over the on-demand and reserved occurrences is that, by permitting consumers to flexibly set their highest bid costs, consumers can save cost in several conditions.

B. Cloud Based TESTBED Mediator

We show the key idea presented in this paper, an mediator-based Cloud testbed (Fig. 1) was designed and implemented using Java and the Java mediator development framework. In a Cloud market, there are many customers and service suppliers, and thus, the Cloud testbed has customer mediators and supplier mediators acting on behalf of customers and suppliers. The responsibilities of each component in the testbed are summarized in Table I.

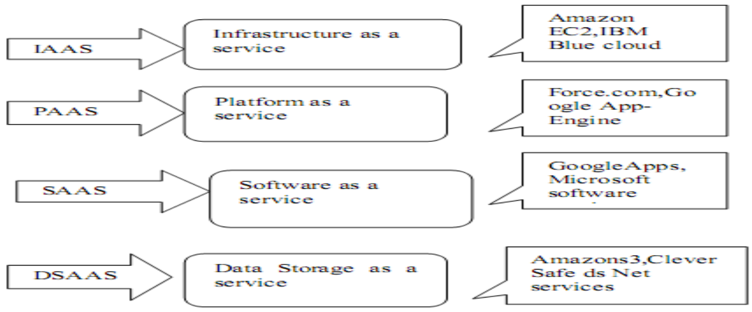

Fig. 1 Mediator-Based Cloud Testbed.

Cloud service suppliers and customers participate in the Cloud market of the testbed through the Cloud market registry. All mediators contributing in the Cloud market is recorded in the Cloud market registry implemented using the Java Mediator Development Framework (JADE) directory facilitator. All customer mediators connected to the Cloud market registry can then recognize and converse with each supplier mediator. Supplier mediators and customer mediators produce service descriptions and specify their preferences with regard to service name, cost, time slot, and negotiation strategy based on a *GUI.*

Table I: Components of the Mediator-Based Cloud Testbed.

Component	Roles
Supplier mediator	Service supplier, service advertisement, time-slot and price negotiation
Customer mediator	Service consumer, service discovery, time-slot and price negotiation
Cloud market registry	Mediator information repository
Cloud simulation controller	Simulation controller for periodic simulation
Cloud status recorder	Status recorder of information of the Cloud market and negotiation outcomes from all negotiation sessions in the market

C. Time Slot and Cost Negotiations

This work considers bilateral negotiation between a customer and a supplier, where both mediators are sensitive to time and adopt a time-dependent concession-making strategy for *TSCNs*. Since both mediators negotiate on both Time Slot and Cost, generating a counterproposal can be making either a concession or a tradeoff between Time Slot and Cost. Hence, a mediator's strategy for many issue negotiation is implemented using both the following:

1) Tradeoff Algorithm: The novelty of this work is adopting a latest tradeoff algorithm, called a "burst method" suggestion, which is planned to improve equally the negotiation velocity and the collective effectiveness. In the previous literature is a multi-issue proposal from mediator.

2) Concession-Making scheme: The concession-making scheme chooses the quantity of concession.

Experimental Result:

In this part, we show the result which we are implemented in Section –III, we will provide result topic-by topic with help of screen shot and table.

A. Cloud Based Testbed Mediator:

In this section, we show that" how message passing between customer service supplier mediators in fig-2. The Cloud status recorder records all negotiation and reservation results of all supplier and customer mediators in the Cloud market. It automatically generates random values for a resource reservation and transmits the values For Cloud service booking; the testbed provides simple service discovery functionality through message passing. A client mediator broadcasts a message indicating the name of the Cloud service that the customer needs to all supplier mediators in and a supplier mediator who has the service replies to the customer in fig-3. In addition, the testbed provides a *TSCN* mechanism to search for a mutually acceptable agreement for leasing the service. In the testbed, supplier mediator represent reserved time slot by marking the memory array and transmit reservation outcomes to the contracted customer to prevent duplicated booking.

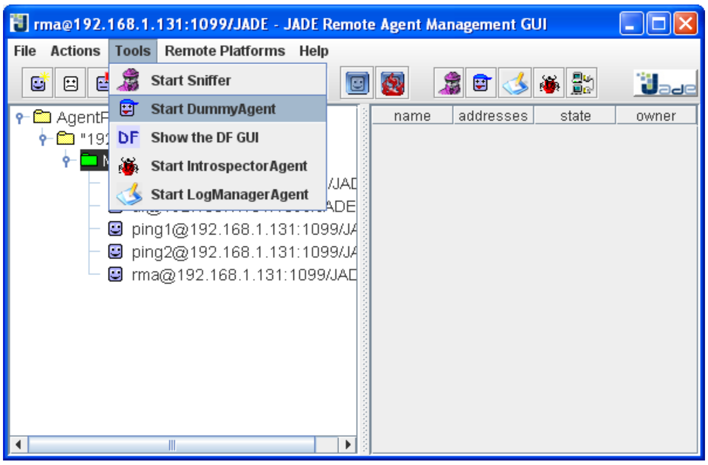

Fig.2. Establish the communication between client and mediators.

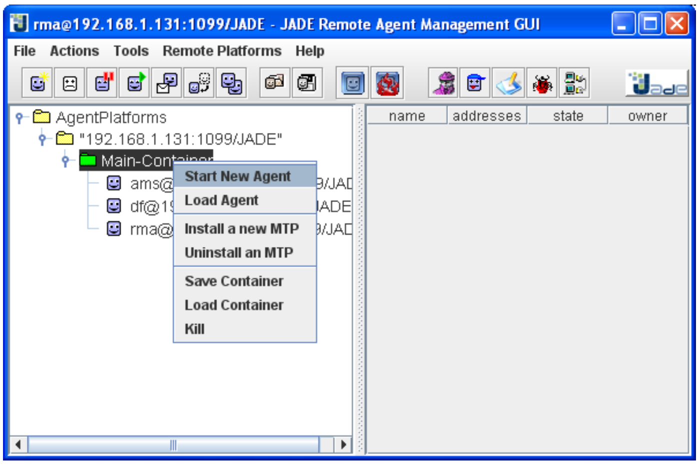

Fig.3. Establish the communication other mediator at a same time.

A. Cloud Pricing through Negotiation:

In this module, we show the cloud registry to customer where customer can find service supplier details and infrastructure details such hardware, software, feedback. This registry is also supportive for service supplier to get knowledge about customer history, requirement details, and payment delivery feedback from other service supplier. Here table II show the service supplier features and table –III shows cloud details who already participated in cloud market.

Table-II Cloud Registry Data.

Input Data	Possible Values	Settings
Cloud Loading (CL)	CL=N_{res} / N_{tot}	0≤CL≤1
No. of Supplier Mediators	Integer	100 service supplier mediators
No. of Customer Mediators	Integer	200 customer mediators
Cloud Services a Supplier leases	String	200 services/supplier (randomly selected)
No. of Negotiation Sessions per each Simulation	Integer	300 negotiation sessions

Table-III Cloud Participants' Inputs for Service Booking.

Input Data	Possible Values	Settings	
		Customer	Supplier
Initial Price (IP)	Integer (Cloud $)	10-60	200-250
Reserve Price(RP)	Integer (Cloud $)	200-250	10-60
First time slot(FT)	Integer, FT<LT	10-60	10-60
Last time slot (LT)	Integer, FT<LT	300-350	300-350
Job size	Integer (Unit: Cloud time)	2-8	2-8
Negotiation Strategy (λ)	Conciliatory(λ<1) Linear (λ=1) Conservative(λ>1)	1/3, 1.0, 3.0	1/3, 1.0, 3.0

Negotiation deadline (τ)	Integer (Unit: Round)	50 rounds, 200 rounds	50 rounds, 200 rounds

Cloud market is parameters for the Cloud simulation controller. In the experiment, 100 service supplier mediators and 200 customer mediators registered in the Cloud market. They are automatically produced by the controller, and the controller randomly invokes a customer mediator 300 times for each simulation to start a service reservation. The selection of the value of each variable is supported on experimental tuning. Experimental tuning shows that 300 negotiation session for each simulation are enough to obtain stable results.

A. Time Slot-Cost Negations

In this module, we demonstrate the application of the *TSCN* mechanism for cost Cloud resources. Here, we show that "how will work our proposed time-cost negation scheme in table". We also current feature of this how the customer are booking the according their time and getting service with his/her own time with affordable cost. We implemented a Cloud computing testbed for simulating an Infrastructure as a Service Cloud supplier using the Xen hypervisor that virtualizes computer hardware to many guest operating systems. The testbed consists of the following: 1) an *IaaS* Cloud supplier represented by a supplier mediator and 2) several customers represented by customer mediators. Customer and supplier mediators can adopt one of the four pricing models: on demand, reserved, spot, or *TSCN* instances.

Table-V: Hourly Usage Cost of Four Pricing Models.

	On-demand	Reserved	Spot	PTN		
					IP	RP
[Small instance] vCPU:4 Memory:600MB Storage: 40GB	$0.085	$0.030	$0.010 - $0.100	Supplier	$0.100	$0.010
				Customer	$0.010	$0.100
[Large instance] vCPU:8 Memory: 1GB Storage: 80GB	$0.340	$0.120	$0.040 - $0.400	Supplier	$0.400	$0.040
				Customer	$0.040	$0.400

The hardware source of the Cloud is virtualized by a Xen hypervisor (version 4.0.1) that launches two guest domain kinds (Domain 0 and Domain U).Whereas Domain 0 is the privileged guest running on the hypervisor with straight hardware access and guest management responsibilities, Domain U (Domain 1-N) launched and controlled by Domain 0 is the unprivileged guest with no straight access to the hardware. Domain U is gives to customers as *VM* instances. In Domain 0, the supplier mediator is implemented using JADE. The Cloud gives two types of instances, namely, small instances and huge instances, and their hourly costs are listed in Table V. The cost for both on-demand and reserved instances follow that in Amazon EC2.

V. Conclusion

In this paper, we designs and implements a *TSCN* mechanism for Cloud service booking. *TSCN* mechanism is designed for both Time Slot and Cost negotiations. Although there are single-issue negotiation mechanism and many-issue negotiation method for Grid resource negotiation, not any of these efforts believes time-slot negotiation. We current new tradeoff method, known as the "burst method" suggestion, has been planned to progress equally the negotiation velocity and the collective effectiveness of Time Slot and Cost in a many issue negotiation. *TSCN* mediators can concurrently construct several suggestions that manufacture the same collective effectiveness other than diverge only in terms of the personality Time Slot and Cost utilities. We have developed a mediator-based Cloud testbed for design the TSCN mechanism.

For future work, this paper can be utilized for following works1) considering and specifying other negotiation issues (e.g., QoS); 2) enhancing the planned tradeoff mechanism by adaptively controlling the quantity of concurrent suggestions in a burst method suggestion to decrease the computational complexity; and 3) providing a user interface for translating high-level user preferences into low-level technological specifications of the time-slot function.

References

1. R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Gener. Comput. Syst.*, vol. 25, no. 6, pp. 599–616, Jun. 2009.
2. A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, T. Nakata, J. Pruyne, J. Rofrano, S. Tuecke, and M. Xu, "Web Services Agreement Specification (WS-Agreement)," WS-Agreement Specification from the Open Grid Forum (OGF), Mar. 2007. [Online]. Available: http://ogf.org/documents/GFD.107.pdf
3. J. Wilkes, "Utility functions, costs, and negotiation," in *Market Oriented Grid and Utility Computing*, R. Buyya and K. Bobendorfer, Eds. New York: Wiley, 2009.

4. I. Foster, C. Kesselman, C. Lee, B. Lindell, K. Nahrstedt, and A. Roy, ,"A distributed resource management architecture that supports advance, booking and co-allocation," in *Proc. 7th IWQoS*, Los Alamitos, CA, Mar. 1999, pp. 27–36.
5. I. Foster, A. Roy, and V. Sander, "A quality of service architecture that combines resource reservation and application adaptation," in *Proc. 8th Int. Workshop Quality Service*, 2000, pp. 181–188.
6. M. A. Netto, K. Bubendorfer, and R. Buyya, "SLA-based advance booking with flexible and adaptive time QoS parameters," in *Proc. 5th Int. Conf. Service-Oriented Comput.*, Vienna, Austria, Sep. 2007, pp. 119–131.
7. S. Venugopal, X. Chu, and R. Buyya, "A negotiation mechanism for advance resource reservation using the alternate offers protocol," in *Proc. 16th IWQoS*, Twente, The Netherlands, Jun. 2008, pp. 40–49.
8. K. M. Sim, "Grid resource negotiation: Survey and new directions," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 40, no. 3, pp. 245–257, May 2010.
9. J. Yan, R. Kowalczyk, J. Lin, M. B. Chhetri, S. K. Goh, and J. Zhang, "Autonomous service level agreement negotiation for service composition provision," *Future Gener. Comput. Syst.*, vol. 23, no. 6, pp. 748–759, Jul. 2007.
10. F. Lang, "Developing dynamic strategies for multi-issue automated contracting in the mediator based commercial grid," in *Proc. IEEE Int. Symp. CCGrid, Workshop Mediator-Based Grid Econ.*, Cardiff, U.K., May 2005, pp. 342–349.
11. K. M. Sim, "Towards complex negotiation for Cloud economy (position paper)," in *Proc. 5th Int. Conf. Grid Pervasive Comput.*, *Lecture Notes in Computer Science*, 2010, pp. 395–406.
12. K. M. Sim, "Equilibria, prudent compromises, and the "waiting" game," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 4, pp. 712–724, Aug. 2005.
13. A. Rubinstein, "Perfect equilibrium in a bargaining model," *Econometrica*, vol. 50, no. 1, pp. 97–109, Jan. 1982.
14. H. Gimpel, H. Ludwig, A. Dan, and R. Kearney, "PANDA: Specifying policies for automated negotiations of service contracts," in *Proc. ICSOC*, vol. 2910, *LNCS*, New York, 2003, pp. 287–302.
15. R. Lawley, M. Luck, K. Decker, T. R. Payne, and L. Moreau, "Automated negotiation between publishers and clients of grid notifications," *Parallel Process. Lett.*, vol. 13, no. 4, pp. 537–548, Dec. 2003.
16. K. M. Sim, "G-commerce, market-driven G-negotiation mediators and Grid resource management," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 6, pp. 1381–1394, Dec. 2006.