Design Considerations to Realize Automated SLA Negotiations in a Multi-Cloud Brokerage System

Seokho Son[1], Dong-Jae Kang[2], Jin-Mee Kim[3]
Cloud Computing Research Department[1,2], Basic Research Center for Software[3]
Electronics and Telecommunications Research Institute
Daejeon, Republic of Korea
{shsonkorea[1], djkang[2], jinmee[3]}@etri.re.kr

***Abstract*—Cloud broker concept is considered as one of solutions for federating distributed multi-Cloud. In order to design and realize a Cloud brokerage system, we need to devise establishment and management of SLA (Service Level Agreement). Whereas a SLA negotiation in a Cloud is assumed to be a basic functionality to establish a SLA, there has been some lack of efforts to apply, configure, and design SLA negotiation mechanism for a Cloud broker, which arbitrates multiple Cloud providers. In this paper, therefore, we design a multi-Cloud broker and reveal design considerations to realize an automated SLA negotiation in a multi-Cloud broker. Briefly, the decision issues to realize a SLA negotiation are as follows: 1) Negotiation lifecycles according to types of Cloud broker, 2) negotiation protocol, 3) negotiable SLA issues, 4) multi-issue negotiation support, and 5) deployment position of negotiation agents.**

Keywords—Cloud Computing; Multi-Cloud; Cloud Broker; SLA Management; SLA Negotiation

I. INTRODUCTION

The number of Cloud providers has increased worldwide and provide various service types under various resource configurations and user-interfaces. So, Cloud service consumers (CSC) are becoming hard to find the most suitable service from a Cloud service provider (CSP). To help CSCs from this complex situation, Cloud computing environment needs to support an intermediary that provides consumers a unified interface for using various Clouds. As an effort for such an intermediary, a Cloud service broker (CSB) has recently emerged as a promising concept. NIST defined general abilities of a cloud broker as 1) service intermediation, 2) service aggregation, and 3) service arbitrage [1]. Therefore, to CSPs (a consumer), a CSB can act on behalf of a CSC (providers, respectively). In a Cloud computing management platform, SLA establishment and management are essential because the provisioning unit in a Cloud is a service. A CSP and a CSC can make a contract using a SLA. As participants in a Cloud may be independent bodies, in order to establish a SLA, some mechanisms must be in place to resolve the different preferences of those entities. A negotiation mechanism is effective in resolving those different preferences.

Since a CSB is an intermediary between a CSC and CSP, to design and realize a Cloud brokerage system, we need to devise establishment and management of SLA. Whereas a SLA negotiation in a Cloud is assumed to be a basic functionality to establish a SLA, there has been some lack of efforts to realize an automated SLA negotiation mechanism in a Cloud broker, which arbitrates multiple Cloud providers.

There are several researches that provide Cloud brokerage architecture. R. Buyya, R. Ranjan, and R. N. Calheiros presented the architecture of a federated Cloud computing environment to support the scaling of applications across multiple Clouds. Whereas the proposed architecture includes a Cloud broker that supports a SLA negotiation, a negotiation mechanism is not specified in [2]. F. Jrad, J. Tao, and A. Streit proposed a Cloud broker operating in an Intercloud environment, and the proposed broker is focused on finding the most suitable Cloud resources taking into account of user requirements specified by SLA [3]. Whereas the proposed broker includes a simple process flow for a SLA negotiation to find a matching result, [3] does not provide a detailed configurations and parameters for a negotiation mechanism.

WS-Agreement specification is one of well-known standards introduced by the Open Grid Forum (OGF) for a Web Services protocol for establishing agreement between two parties, and WS-Agreement Negotiation Protocol (WSAN) is an extension of WS-Agreement [4]. WSAN is a protocol for negotiating SLAs of web services between two parties. WSAN provides a generalized negotiation framework. It is hard for a brokerage system designer to simply apply. S. Son and K. M. Sim specified some negotiation parameters to establish Cloud SLA, but [5] does not provide design considerations in a Cloud brokerage system.

In this paper, we introduce the ongoing design of a multi-Cloud broker named AnyBroker and reveal design considerations to realize an automated SLA negotiation in the multi-Cloud broker. Briefly, the decision issues to realize a SLA negotiation are as follows. 1) Negotiation lifecycles according to types of Cloud broker, 2) Negotiation protocol, 3) Negotiable SLA issues, 4) Multi-issue Negotiation support, 5) Deployment position of negotiation agents. In Section 2, we introduce an ongoing design of a Cloud brokerage system. Section 3 lists design considerations for an automated SLA negotiation mechanism for a Cloud broker. Finally, Section 4 concludes and includes a discussion of future research.

This work was supported by the ICT R&D program of MSIP/IITP, Republic of Korea. [14-000-05-001, Smart Networking Core Technology Development]

COLLABORATECOM 2014, October 22-25, Miami, United States

DOI 10.4108/icst.collaboratecom.2014.257551

II. DESIGN OF A CLOUD BROKERAGE SYSTEM

This Section introduces an ongoing development of a Cloud brokerage system (AnyBroker) that can provide Cloud and network services from multiple Cloud and network service providers to CSCs. Fig. 1 shows the design of AnyBroker architecture. The AnyBroker consists of eight components: 1) web portals, 2) administrative Information management, 3) Business Assistance, 4) Cloud arbitration, 5) optimal service placement, 6) service lifecycle management, 7) service monitoring, and 8) multi-Cloud access management.

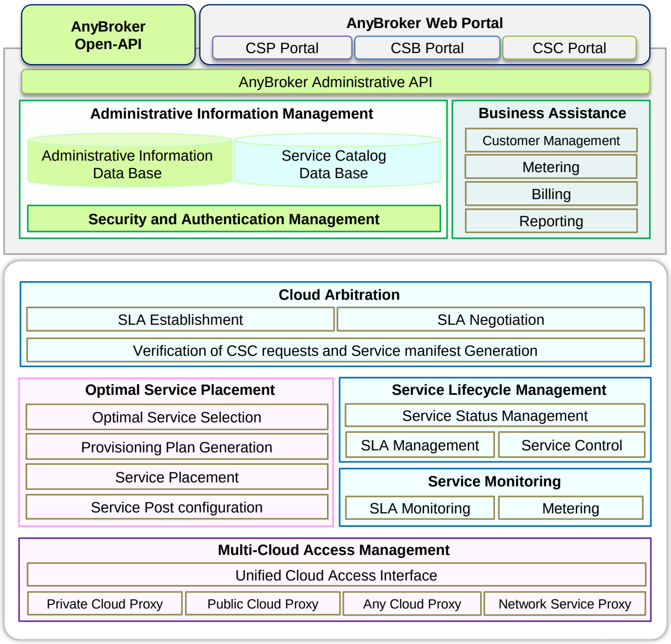

AnyBroker Open-API
AnyBroker Web Portal
CSP Portal
CSB Portal
CSC Portal
AnyBroker Administrative API
Administrative Information Management
Administrative Information Data Base
Service Catalog Data Base
Security and Authentication Management
Business Assistance
Customer Management
Metering
Billing
Reporting
Cloud Arbitration
SLA Establishment
SLA Negotiation
Verification of CSC requests and Service manifest Generation
Optimal Service Placement
Optimal Service Selection
Provisioning Plan Generation
Service Placement
Service Post configuration
Service Lifecycle Management
Service Status Management
SLA Management
Service Control
Service Monitoring
SLA Monitoring
Metering
Multi-Cloud Access Management
Unified Cloud Access Interface
Private Cloud Proxy
Public Cloud Proxy
Any Cloud Proxy
Network Service Proxy

Fig. 1. Architecture of the AnyBroker.

AnyBroker is under development based on CompatibleOne, which is an open source software project for a cloud broker. CompatibleOne is interoperable with various Cloud management platforms (OpenNebula, OpenStack, VMware vCloud), public cloud service providers (Amazon, Azure, CloudSigma, Dimension Data, Go Grid, HPCloud, Joyent, OnApp, RackSpace, Softlayer) [6]. CompatibleOne is used as the multi-Cloud access management component in AnyBroker so that AnyBroker is interoperable with various Clouds. For Cloud arbitration, AnyBroker includes SLA negotiation as a key component.

III. DESIGN CONSIDERATIONS IN SLA NEGOTIATION

A negotiation mechanism consists of the negotiation protocol, negotiation strategy, and utility functions (Fig. 2). The negotiation protocol is a set of rules on communication (e.g., possible actions, language, and utterance turn) for negotiations among involved parties. In a multi-rounds negotiation protocol, counter-proposals can be generated according to the negotiation strategy, which consists of a concession algorithm and tradeoff algorithm. When an agent generates a counter-proposal, the agent needs to concede a proposal because it is hard to reach an agreement without a concession. A concession algorithm determines the amount of concession for each negotiation round, and a tradeoff algorithm is required to generate a proposal in multi-issue (i.e., multi-attributes) negotiation. In multi-issue negotiation, there are multiple issues to negotiate, and the tradeoff algorithm generates a proposal by combining the proposals for individual issues. While we implement an automated SLA negotiation mechanism into the proposed Cloud brokerage system, we faced several design considerations that helps other researchers as follows.

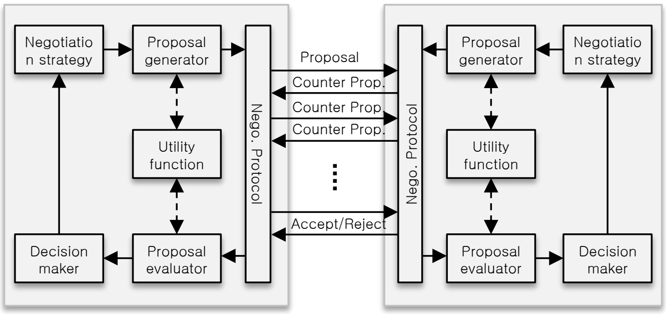


Fig. 2. A General Negotiation Mechanism.

A. *Negotiation lifecycles according to broker types*

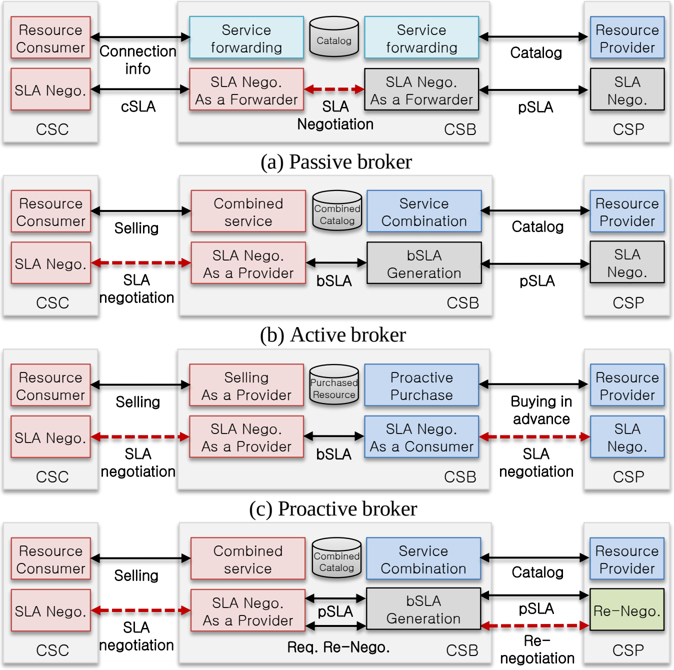
In this Section, we classified a Cloud broker into four types according to profit models as 1) passive broker, 2) active broker, 3) proactive broker, and 4) reactive broker. According to the broker types, SLA negotiation lifecycles can be differentiated. Fig. 3 shows a SLA and service provisioning structure according to broker types. In Fig. 3, pSLA, bSLA, and cSLA are a SLA preference generated by a provider (CSP), a broker (CSB), and a consumer (CSC), respectively. The designer for a Cloud brokerage system can select a brokerage type and

Passive brokerage: passively interconnects a CSP and CSC so that a broker can facilitate Cloud usages. This type connects does not have a profit model. The objective of this broker is to facilitate Cloud usages. Fig. 3(a) shows how the SLA negotiation lifecycle works. Since the passive broker is not interested with making a profit, the broker just help SLA negotiation between a CSP and CSC. So, a CSP sends pSLA to CSB, and a CSC sends cSLA to CSB. Using the pSLA and cSLA, CSB executes SLA negation and intermediate differences between the CSC and CSP.

Active brokerage: reconfigures CSPs' services and acts as a new CSP to CSCs. Active broker make profits by taking commission from CSPs. Fig. 3(b) shows the SLA negotiation lifecycle of the active broker type. CSB negotiates with a CSC by using a modified CSP's SLA (i.e., SLA negotiation using cSLA and bSLA modified from pSLA).

Proactive brokerage: purchases resources or services from CSPs in advance (e.g., purchasing one year reserved instance in a cheap price and selling the instance in on-demand service in a higher price). To a CSP, CSB acts as CSC. On the contrary, CSB acts as a CSP to a CSC. Active broker make profits by making own marketing strategy. In Fig. 3(c) for the proactive broker type, CSB negotiates with a CSP; CSB generates bSLA based on CSB's strategy; finally, CSB negotiates SLA with a CSC.

Reactive brokerage: supports renegotiation between CSPs and CSB when the antecedent negotiation between CSB and a CSC failed. This brokerage type can be combined with the other broker types (e.g., passive and reactive brokerage).



(a) Passive broker

(b) Active broker

(c) Proactive broker

(d) Reactive broker

Fig. 3. Classifiation of Cloud Broker and SLA Negotiatoin Lifecycles.

B. *Negotiation protocol*

A design issue in the negotiation protocol for is the support of a multi-round negotiation and a negotiation deadline (i.e., maximum rounds to bear). If a negotiator has a longer deadline than his opponent, the negotiator would have a strong bargaining position. The designer of a brokerage system needs to determine whether the negotiation rounds and deadline concept are included in the SLA negotiation. Including the negotiation deadline will support advanced bargaining ability, but the deadline will increase negotiation failures and lead to overhead in the speed of SLA establishment. For a multi-round negotiation, Rubinstein's alternating offers [7], which lets agents make counter-offers to their opponents in alternate rounds can be utilized.

C. *Negotiable SLA issues*

A Cloud brokerage system can share negotiable SLA issues (attributes) used in general Cloud management platforms (e.g., availability, response time, and reliability). The designer needs to consider whether the Cloud brokerage system support a negotiation for the price issue. The price negotiation is special because price issue fits a bargaining issue. If price negotiation is not included in a SLA negotiation, the role of the negotiation can be limited to a searching algorithm to find a solution space.

D. *Multi-issue Negotiation support*

The designer of a brokerage system needs to consider multi-issue negotiation support, which resolves a trade-off relationship among multiple SLA issues concurrently. Otherwise, the designer can select a mechanism with consecutive single-issue negotiations when there are multiple SLA issues to negotiate. If the designer determines to support multi-issue negotiation, the designer needs to devise a trade-off algorithm for decision-makings of negotiation agents.

E. *Deployment position of negotiation agents*

In general, negotiation parameters (e.g., negotiation deadline, strategy) of a negotiator should be concealed to the opponent. So, the communication message between negotiators would be their proposals and their decisions have to be made by them self. However, user-interface of a CSB is a service web portal in general. If the CSB makes consumers fill in negotiation parameters in the service web portal, consumers may concern that their negotiation strategy would be revealed to the opponent, who is a broker on behalf of sellers. Therefore, a designer of a broker needs to determine the deployment position of negotiation agents. To prevent a leak to providers about the negotiation strategy of a consumer, negotiation agent of a consumer should be deployed in the consumer's site. If the negotiation agent is deployed in the broker system, this policy has to be agreed with consumers so that the consumers trust that the broker would not cheat SLA negotiation.

IV. CONCLUSION AND FUTURE WORK

In this paper, we design architecture of a multi-Cloud broker and reveal design considerations to realize an automated SLA negotiation in a multi-Cloud broker. Whereas there are several Cloud brokerage architectures and SLA negotiation frameworks, it is not sufficient to realize a practical SLA negotiation. Therefore, the significance of the paper is that this paper can help a designer for a Cloud brokerage architecture make decisions in realizing a SLA negotiation. Finally, the authors expect this work can be extended in two ways: 1) considering and specifying additional negotiation issues in Cloud SLAs and 2) optimizing negotiation parameters for increasing both negotiation success rate and resource utilization of Clouds.

REFERENCES

[1] M. Hogan, F. Liu, A. Sokol, and J. Tong, "Nist clound computing standards roadmap," *NIST Special Publication 500-291*, July 2011.

[2] R. Buyya, R. Ranjan, and R. N. Calheiros, "InterCloud: Utility-oriented federation of cloud computing environments for scaling of application services," in *Proc. the 10th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP'10)*, Busan, pp. 13–31, May 2010.

[3] F. Jrad, J. Tao, and A. Streit, "SLA based service brokering in intercloud environments," in *Proc. the International Conference on Cloud Computing and Services Science (CLOSER 2012)*, Portugal, pp. 76–81, April 2012.

[4] OGF, "WS-Agreement Negotiation Specification v1.0," OGF, 2011.

[5] S. Son and K. M. Sim, "A price and time slot negotiation mechanism for cloud service reservations," *IEEE Trans. on Systems, Man and Cybernetics, B: Cybernetics*, vol. 42, no. 3, pp. 713–728, June 2012.

[6] S. Yangui, I.J. Marshall, J.P. Laisne, S. Tata, "CompatibleOne: The open source cloud broker," *Journal of Grid Computing*, vol. 12(1), pp. 93–109, March 2014.

[7] A. Rubinstein, "Perfect equilibrium in a bargaining model," *Econometrica*, vol. 50, no. 1, pp. 97–109, 1982.