A Cloud Service Broker for SLA-based SaaS Provisioning

Elarbi Badidi
College of Information Technology
United Arab Emirates University
Al-Ain, United Arab Emirates
Email: ebadidi@uaeu.ac.ae

Abstract— **The growing adoption of cloud computing and the proliferation of Internet-enabled handheld devices are changing the services landscape. Given the abundance and the variety of Software-as-a-Service (SaaS) offerings, we propose, in this paper, a framework for SaaS provisioning, which relies on brokered Service Level agreements (SLAs), between service consumers and SaaS providers. A Cloud Service Broker (CSB) helps consumers selecting the right SaaS provider that can fulfill their functional and quality-of-service (QoS) requirements. Its Selection Manager component ranks SaaS providers by matching their QoS offerings against the QoS requirements of the service consumer. Furthermore, the CSB is in charge of negotiating the SLA terms – using a multi-attributes negotiation model -- with a selected SaaS provider on behalf of the service consumer, and monitoring the compliance to the SLA during its implementation.**

Keywords-cloud computing; cloud services; quality-of-service; service level agreement, service level objective

I. Introduction

Cloud computing has revolutionized the standard model of service provisioning, allowing delivery over the Internet of virtualized services that can scale up and down in terms of processing power and storage. The three main models of cloud services are: *Infrastructure-as-a-Service (IaaS)*, *Platform-as-a-Service (PaaS)*, and *Software-as-a-Service (SaaS)* [1].

The success of cloud services depends enormously on the level of satisfaction of cloud customers in terms of performance and quality-of-service (QoS) they get from cloud service providers. QoS refers to a set of qualities or characteristics of a service, such as *availability*, *security*, *response-time*, *throughput*, *latency*, *reliability*, and *reputation*. Such qualities are of interest for service providers and service consumers alike. They are of interest for service providers when implementing multiple service levels and priority-based admission mechanisms. The agreement between the customer and the service provider, known as the *Service Level Agreement (SLA)*, describes agreed service functionality, cost, and qualities [2].

As of today, IaaS and PaaS cloud service delivery models are using SLA-based service provisioning. However, its adoption in the SaaS service delivery model is still at its beginning. Cloud service providers typically use a resources' over-provision strategy in their effort to meet the SLA requirements from different customers, in terms of availability and performance [3]. They allocate resources to customers based on their requirements in the worst-case scenario. As a result of this policy, cloud provider servers may be sub-optimally used as some allocated resources may be idle at run-time. As the number of cloud customers is growing significantly, it is essential for cloud service providers to overcome this situation by being able to allocate resources on an as-needed basis. Thus, they should have the means to monitor resources' usage and evaluate various QoS metrics to be able to honor their SLA commitments.

Given the limited number of solutions on SLA-based SaaS provisioning and the heterogeneity of SaaS providers' APIs, we set out to develop a framework that will: (1) allow service consumers to express their functional and nonfunctional requirements, (2) hide from service consumers the heterogeneity of SaaS providers' APIs, (3) allow implementing QoS-driven selection of SaaS providers, (4) allow implementing SLA negotiation, and (5) permit monitoring and assessment of SLAs implementation. The main component of our proposed framework is the Cloud Service Broker (CSB), which is in charge of mediating between service consumers and SaaS providers in order to reach agreements that explicitly describe expected service functionality and QoS levels. Besides, The CSB carries out other activities such as selection of SaaS providers, SLA negotiation, and SLA compliance monitoring.

The rest of this paper is organized as follows. The next section describes background information on the concepts of SLA, Service Level Objectives (SLOs), and cloud services. Section 3 describes related work. Section 4 presents an overview of the proposed framework. Section 5 describes the SLA negotiation process. Section 6 describes the multi-attributes negotiation model. Finally, Section 7 concludes the paper and describes future work.

II. Background

A. SLAs and SLOs

A SLA is an agreement regarding the guarantees of a service. It defines mutual understandings and expectations of a service between the service provider and service consumers. It consists of sections describing the commitments to service quality and service levels that the service provider has to

978-1-908320-13/1/$25.00©2013 IEEE

guarantee. The service guarantees concern the operations to be executed and the QoS to be delivered. Table 1 depicts the typical sections of a SLA [4].

Sun et al. [5] decomposed an SLA life cycle into five phases, which are: Development, Negotiation and Sales, Implementation, Execution, and Assessment. Several research works have undertaken substantial efforts to standardize the use of SLAs. The most significant initiatives are the WSLA framework [2], the WS-Agreement specification [6], and the SLAng specification Language [7].

Service level is a performance measure of how well the service provider is responding to incoming service requests. SLOs are the goals of the service provider, such as the percentage of service request the service provider wants handled within a certain number of seconds (e.g., 90% within 10 seconds). They represent a commitment of the service provider to maintain a certain level of service in a predefined period [8]. SLOs may vary considerably depending on the services offered. In other words, SLOs will be specific to business needs. Service providers have also to take into account how their main competitors are performing. They have to set their objectives to match or perform better than their competitors. Business and customer needs, as well as the competitive landscape, will govern service level objectives of each service provider. A cloud service provider may offer different service levels for the same service as a result of deploying the same service on different data centers with various capabilities. The levels would differ in terms QoS and cost. A typical SLA may include the following SLOs: *Service Availability, Service Response Time, Service Availability, Service Outage Resolution Time, Failover Window For Disaster Recovery, Maintenance Notification, Proactive Service Outage Notification, and Reason for Outage* [9].

B. *Cloud Services*

Cloud services are applications or services offered by means of cloud computing. Typical examples of cloud services include desktop office applications (word processing, spreadsheets, and presentations). Google Docs and Microsoft office Web Apps are cloud services of this category. Other examples include storage services, calendar, notebooks' applications, and many more.

TABLE 1- MAIN SECTIONS OF A SLA

Purpose	Describes the motives behind the creation of the SLA
Parties	Represents the parties involved in the SLA and their respective roles (provider and consumer).
Validity period	Defines the period during which the SLA will be valid.
Scope	Defines the services covered in the agreement.
Restrictions	Define the necessary steps to be taken in order for the requested service levels to be provided.
Service-level objectives	Represent the levels of service that both the service consumer and the service provider agree on. They typically include a set of QoS indicators.
Penalties	Specify the penalties for not meeting the stated SLOs, such as getting discounts or having the right to terminate the contract.
Exclusions	Specify what is not covered in the SLA.
Administration	Defines the processes to assess the SLA objectives, and describes the responsibility of the service provider regarding the control of each of these processes.

Nearly, all large software corporations, such as Google, Microsoft, Amazon, IBM, and Oracle, are providing various kinds of cloud services. Besides, many small businesses have launched their own Web-based services, mainly to take advantage of the collaborative nature of cloud services. As we mentioned earlier in the introduction, cloud services models are IaaS, PaaS, and SaaS. The user of a cloud service has access to the service through a Web interface or via an API.

One of the underlying advantages of deploying services on the cloud is the economy of scale. Using the cloud infrastructure, a service provider can offer better, cheaper, and more reliable services than is possible within its premises. The cloud service can utilize the entire processing and storage resources of the cloud infrastructure if needed. Another advantage is scalability in terms of computing resources. Service providers can scale up when the demand for service rises significantly. Conversely, they can scale down when the demand for service decreases. Another benefit is enabling service consumers to use services on a pay-as-you-go or a monthly subscription basis.

SaaS represents the trend of the future and the most common form of cloud service development and deployment. With SaaS, providers deploy their software over the Internet and deliver it to thousands of customers. Using this delivery model, the SaaS provider may license its service to customers through a subscription or a pay-as-you-go model.

III. RELATED WORK

Cloud service brokerage and the issues of SLA management -- SLA negotiation in particular -- are the subject of several research efforts over the last few years. Gartner predicts that, as cloud services proliferate*, Cloud Service Brokers* will emerge and will mainly be in charge of the management of the utilization, performance, and delivery of cloud services. They will broker relationships between a service consumer and multiple service providers [10].

In the European project mOSAIC (www.mosaic-cloud.eu), a key component is the Cloud Agency, which aims to allow service consumers to delegate to the agency all SLA management tasks, the monitoring of resource utilization and in some circumstances re-negotiation of SLA terms [11]. The Cloud Agency system depends on several agents that aim to manage cloud resources and services offered by various cloud providers. The SLA@SOI project is an ambitious project whose goals are defining a comprehensive view for SLAs' management and developing a framework for SLA management. The framework can be incorporated into a Service Oriented Infrastructure (SOI) [12]. Theilmann et al. developed, in the context of this project, a reference architecture for multi-Level SLA management [13]. This architecture aims to provide a comprehensive solution for SLA management that can: (1) support SLA management across multiple layers of a SOI, (2) cover the entire SLA and service life cycle, and (3) be used in various use cases. OPTIMIS [14] is a toolkit, which aims at optimizing the entire cloud service life cycle, including service creation, deployment, configuration, and operation, by considering nonfunctional issues like trust, cost, and risk. The toolkit targets mainly service providers and infrastructure providers and may also impact other actors such as brokers, and service consumers. The fundamental process during deployment is the negotiation of SLA terms between service providers and infrastructure providers. Moore et al. [15] designed and implemented a Web service broker to help manage the interactions and data exchange between clients, looking up for services, and SaaS providers, which are interested in promoting their services in a trusted environment. The service broker acts as a container for publishing heterogeneous SaaS applications from different SaaS providers.

In SOA and grid environments, several efforts have investigated the issue of service negotiation and the specification of machine-readable SLAs. The main motivation of these efforts is to guarantee the quality of service specified in the SLA and fair satisfaction of the participants. Dan et al. [2] describe a framework for providing differentiated levels of service to service consumers in a SOA environment by means of SLAs and automated management. The framework encompasses WSLA – for the creation and negotiation of SLAs --, a system for dynamic allocation of resources based on SLOs, and a system to monitor compliance with the SLA. Modica et al. [16] focused on the use of WS-Agreement for the specification of SLAs. They proposed to incorporate new functionality into the interaction protocol to allow the parties of an agreement to renegotiate and modify its terms during the service provision.

A more ambitious goal of several efforts is to support the automated negotiation of the SLA terms. Silaghi et al. [17] introduced a framework for building SLA automatic negotiation strategies under time constraints in computational grids. Resinas et al. [18] tackled the problem of building automated service agreement negotiation systems that rely on a bargaining protocol and work in open environments. They proposed a bargaining architecture that provides support for many requirements, which are missing from other proposals, such as multi-term negotiation, heterogeneity of the parties, management of partial information about the parties, and simultaneous negotiations with different parties. Hasselmeyer et al. [19] described a brokered-based approach to SLA negotiation. Their solution relies on the idea of outsourcing SLA negotiation to third parties (agent brokers) acting on behalf of their clients.

This work shares with these efforts the common goal of mediating between service consumers and service providers and providing support for automated SLA negotiation and management. Finding the right SaaS offering is not an easy task for service consumers given the plethora and the variety of SaaS offerings. Moreover, dealing with a SaaS provider requires knowledge of its operating environment, the availability of management tools, its security levels and data recovery approaches, and the service terms and conditions. Collecting this information for multiple service providers is likely to be a demanding task that is expensive and time consuming. The CSB with its know-how and value-added services will assist service consumers in: (a) finding appropriate SaaS offerings, (b) negotiating SLA terms, (c) monitoring and assessing the implementation of SLAs, and (d) offering a single interface to multiple SaaS providers.

IV. SLA-DRIVEN SAAS SERVICE PROVISIONING FRAMEWORK

Fig. 1 depicts the proposed framework for SLA-based service provisioning. The main components of the framework are: service consumers (SCs), the Cloud Service Broker (CSB), the Monitoring Infrastructure, and SaaS Providers (CSPs).

A. Cloud Service Broker

As we mentioned earlier, the CSB is a mediator service that decouples service consumers from SaaS providers. Given that service consumers do not normally have the capabilities to negotiate, manage, and monitor QoS, they delegate management tasks, such as selection of appropriate SaaS providers and SLA negotiation, to the CSB.

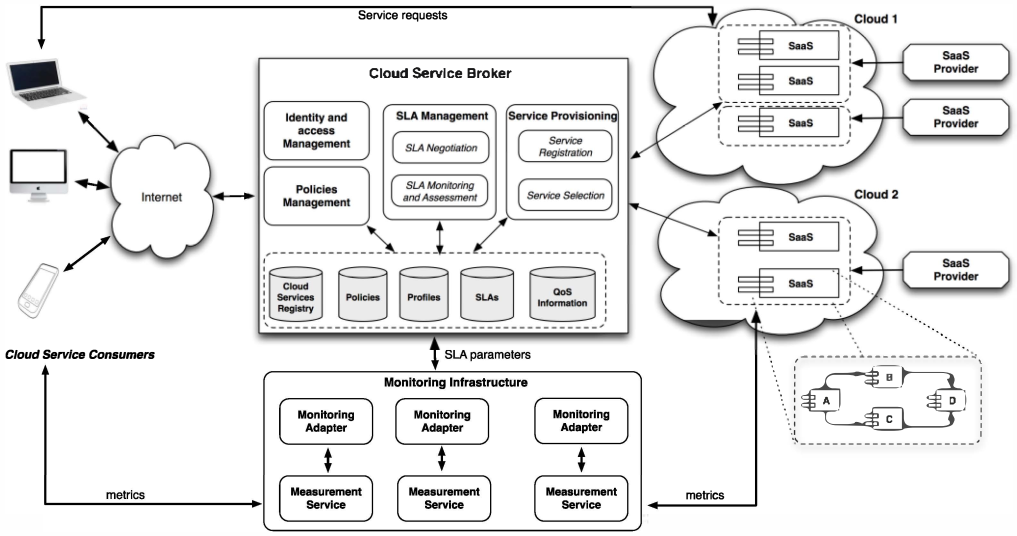

Figure 1. Broker-based Framework for SaaS Provisioning

The architecture of the CSB includes several management operations that cooperate in order to deliver personalized services to clients. These operations are: *Identity and Access Management (IAM)*, *Policies Management (PM)*, *SLA Management (SLAM)*, and *Service Provisioning (SP)*. They are under the control of a *Coordinator* component. The back-end databases maintain information about services' policies, consumers' profiles and preferences, SLAs, and dynamic QoS information.

The *Selection Manager,* which implements SP management operations, is in charge of implementing different policies for the selection of suitable SaaS providers, based on the consumer's QoS requirements and the SaaS providers' QoS offerings. Several works have investigated the issue of cloud services' selection [20] [21]. The *SLA Manager*, which implements SLAM management operations, is in charge of carrying out the SLA negotiation process between a service consumer and a selected SaaS provider. It approaches this SaaS provider to determine whether it can ensure the required level of service given its current conditions. Then, the consumer and the SaaS provider sign a contract. The contract describes the type of service, the service level to ensure, the cost of service, and actions to take in case or repeated violations of the agreement. If the selected SaaS provider is unable to deliver the required level of service, the broker selects another SaaS provider and reiterates the negotiation process. The *Profile Manager*, which implements IAM management operations, is responsible for managing service consumers' profiles, including their preferences in terms of personalized services and required QoS. The *Policy Manager*, which implements PM management operations, is responsible for managing different kinds of policies such as authorization policies and QoS-aware selection policies of service providers.

B. Monitoring Infrastructure

As depicted in fig. 1, the *Monitoring Infrastructure* includes several *Monitoring Adapters* and *Measurement Services. Measurement Services* are new players in the service-provisioning ecosystem. They allow service providers to have independent monitoring of their offerings, and enable service consumers to make sure that they are getting the promised level of service. Examples of measurement and monitoring services are Keynote (keynote.com), Monitis (monitis.com), and Uptrends (uptrends.com). For each Measurement Service there is a *Monitoring Adapter, which* is in charge of mapping resource metrics, measured by the Measurement Service, into SLA parameters, monitoring current service levels and evaluating their compliance with the SLA. Monitoring Adapters may be added to the framework if service consumers and service providers do prefer specific monitoring companies. SLAs are hidden from the measurement services. The adapters use the APIs of the Measurement Services to get current QoS data. For example, Monitis is providing an open API (portal.monitis.com/open-api), which allows accessing most of the commands of Monitis dashboard. Similarly, Keynote is offering an API (https://api.keynote.com/) to access Keynote measurement data.

C. SaaS Providers

As illustrated in Fig. 1, SaaS providers can offer several types of services that they implement using simple or composite Web services. In order to determine their current QoS offering, SaaS providers need to use monitoring techniques that allow collecting measurement data at selected observation points. By aggregating collected data, a SaaS provider can determine the value of each QoS indicator. If there is a significant drop in its current QoS offering, the SaaS provider might add additional resources to meet its SLA commitment. The *SLA Manager* of a SaaS provider is responsible for managing SLA templates, negotiating the SLA terms with the CSB, or directly with service consumers, and implementing SLA and making adjustments.

V. SLA Negotiation

Fig. 2 depicts a scenario of the SLA negotiation process. Fig. 3 and Fig. 4 show the state charts of the CSB's SLA Manager and the SaaS provider's SLA Manager respectively. The steps of the negotiation process are:

1) The service consumer submits an SLA Request to the CSB to find out an appropriate SaaS provider that can meet its service functional and nonfunctional requirements.

2) After authenticating the service consumer, the Coordinator requests its profile from the Profile Manager. Then, it requests from the *Selection Manager* to select a suitable SaaS provider, which can deliver the service according to the service consumer requirements.

3) The Coordinator requests policies of the selected SaaS provider from the Policy Manager.

4) If the service consumer profile is available in the profile repository, for example, because the service consumer had previously used some services of the CSB, the Coordinator may determine whether the SaaS providers, found by the *Selection Manager*, can handle or not the service consumer request. This decision relies on the profile of the service consumer and policies of the selected SaaS providers.

5) If the service consumer's profile is not available in the profile repository, then the Coordinator asks the service consumer to provide information, such as service preferences and desired levels of QoS, in order to create a new profile for the service consumer.

6) If at least one SaaS provider can meet the service consumer requirements, the Coordinator requests from the SLA Manager to negotiate with that SaaS provider the terms and conditions of service delivery.

7) The CSB' SLA Manager forwards the SLA Request, to the SaaS provider's SLA Manager, requesting a proposal from the SaaS provider. The SaaS provider's SLA Manager parses the SLA Request and validates it against its SLA templates.

8) If the SLA Request is acceptable to the SaaS provider, then its SLA Manager responds to the SLA Request by sending back an SLA proposal. The CSB analyses the

SLA proposal to determine whether it meets all functional and nonfunctional requirements of the service consumer.

9) If the service consumer's expectations can be met, then the CSB accepts the offer of the SaaS provider and sends an SLA Confirmation to the SaaS provider. Otherwise, it rejects the offer and makes a counter-proposal with different conditions, terms, costs, etc.

The operations of the SLA Manager of the SaaS provider, with regards to SLA Negotiation, are as follows:

1) After receiving an SLA Request, the SLA Manager analyses the SLA Request.

2) If there is no issue with the structure of the SLA Request, then the SLA Manager validates the SLA Request against available SLA templates.

3) If the SLA Request is acceptable, then the SaaS provider's SLA Manager creates an SLA Proposal and sends it to the CSB. At this point, the SaaS provider will wait for SLA approval or dismissal from the CSB.

4) If the SLA request is not acceptable to the SaaS provider, then its SLA Manager may suggest an alternative SLA to the CSB and waits for acceptance or rejection.

5) Upon reception of the SLA approval from the CSB, the SLA Manager subscribes the service consumer in its registry and the SLA becomes ready for implementation.

In step 8 and step 9 of the above SLA negotiation protocol, a multi-attributes negotiation takes place between the SLA Managers of both parties according to the negotiation model that we describe in the next section.

VI. THE MULTI-ATTRIBUTES NEGOTIATION MODEL

Selection algorithms do not guarantee that the selected SaaS offer is the best one. Therefore, to reach an agreement, the service consumer and the selected service provider need to negotiate the various SLOs of the service. We assume that QoS indicators are in normalized form with values between 0 and 1. A value of 1 means highest quality and 0 means lowest quality. When submitting a SLA request to the CSB, the service consumer specifies her preferences with regards to the normalized QoS indicators that she can tolerate. Let $X = \{X_1, X_2, ..., X_m\}$ be the list of QoS indicators considered in the system.

Let $M = \{min_1, min_2, ..., min_m\}$, $0 \le min_i \le 1$, be the minimum quality requirements that the service consumer tolerates for a certain service. Let $Q = \{q_1, q_2, ..., q_m\}$, $0 \le q_i \le 1$, be the QoS offering of the SaaS provider SP for that service. Therefore, the service consumer and the SaaS provider need to negotiate m QoS parameters. Multi-attributes negotiation is a process in which the two parties negotiate multiple issues concurrently. This is a common problem in business. The parties, very often, recognize the existence of differences of interest over several issues; but, the need for win-win cooperation is the incentive for seeking to reach a compromise agreement [22].

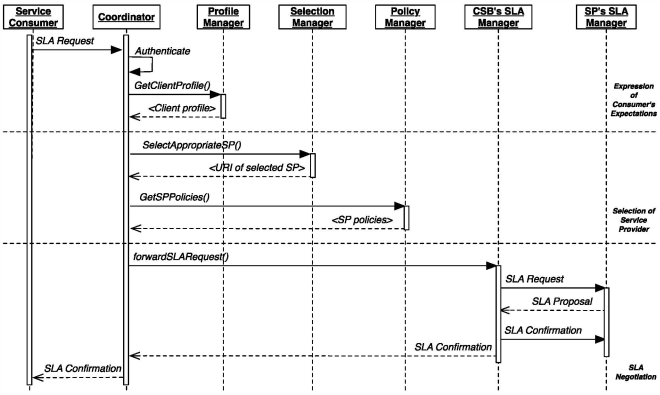

Figure 2. SLA Negotiation Process

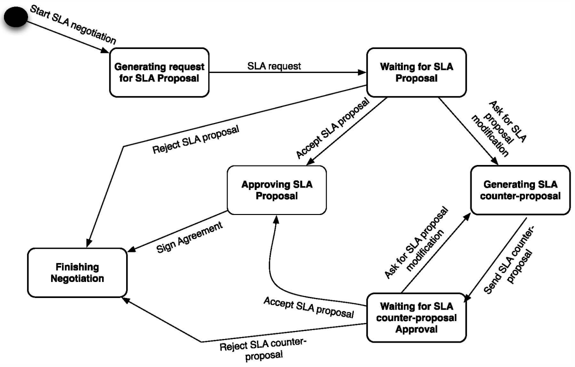
Start SLA negotiation
Generating request for SLA Proposal
SLA request
Waiting for SLA Proposal
Reject SLA proposal
Accept SLA proposal
Ask for SLA proposal modification
Approving SLA Proposal
Generating SLA counter-proposal
Sign Agreement
Finishing Negotiation
Ask for SLA proposal modification
Send SLA counter-proposal
Accept SLA proposal
Waiting for SLA counter-proposal Approval
Reject SLA counter-proposal

Figure 3. State chart of the CSB's SLA Manager

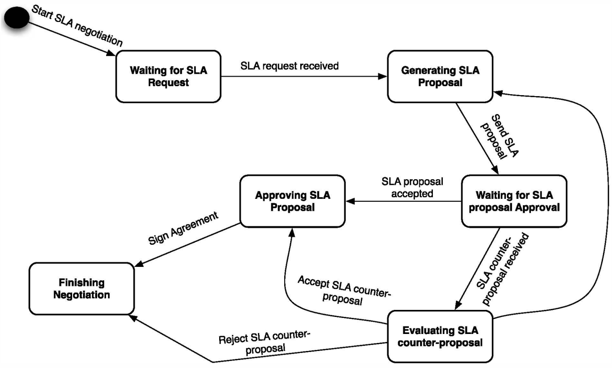

Figure 4. State chart of the SaaS provider's SLA Manager

To reach an agreement, The SLA Managers of both parties have to go through several rounds of negotiation of offers and counter-offers until they reach an agreement or reach a predefined maximum number of rounds. In each round, the CSB's SLA Manager evaluates an aggregate utility function and determines if the offer of the SaaS provider is acceptable or not. If we assume that the QoS attributes are independent, the linear aggregate utility function can be defined by:

$$U = w_1 U_1 + w_2 U_2 + ... + w_m U_m, \text{ with } \sum w_i = 1 \qquad (1)$$

U_i represents the individual utility function associated with the QoS attribute X_i, and w_i is the weight that the service consumer assigns to that attribute. Various functions may be used to express the service consumer utility of an attribute X_i. We adopt the function used in [23] to express the utility function as: $U_i = x_i^{\beta_i}$

β_i is a measure of the service consumer sensitivity to the QoS attribute x_i. When $\beta_i = 0$, the service consumer is indifferent to QoS attribute x_i. When $\beta_i = 1$, the service consumer is moderately sensitive to QoS attribute x_i (the relationship is linear). When $\beta_i > 1$, the service consumer is increasingly sensitive to QoS attribute x_i. As β_i increases, the service consumer is expressing increasing concern about x_i. For $\beta_i < 1$, as β_i decreases to approach 0, the service consumer is expressing increasing indifference to having x_i.

Equation (1) becomes:

$$U=w_1x_1^{\beta_1}+w_2x_2^{\beta_2}+\ldots w_nx_n^{\beta_n} \quad (2)$$

VII. Conclusion and Future Work

Given the wide acceptance of cloud computing technology and the growing number of SaaS offerings, service consumers are increasingly facing the challenge of finding appropriate SaaS providers that can satisfy their functional and nonfunctional requirements.

In this paper, we have presented a framework for SaaS provisioning that relies on a Cloud Service Broker. The CSB is in charge of mediating between service consumers and SaaS providers, selecting appropriate SaaS providers, and negotiating the SLA terms. SLA negotiation involves the negotiation of multiple SLA parameters with both parties trying to maximize their utility function. We have considered a linear aggregate utility function for the service consumer, which allows reaching a single optimal point. Advanced non-linear algorithms, such as the particle swarm optimization, may offer better results and several Pareto optimal points in the search space.

As a future work, we intend to explore the use of non-linear utility functions for the multi-attribute negotiation model and build a prototype of the framework together with some real scenarios for SLA-based SaaS provisioning.

References

[1] P. Mell and T. Grance, "The NIST definition of cloud computing," National Institute of Standards and Technology, Special Publication 800-145, 2011.

[2] A. Dan, D. Davis, R. Kearney, and A. Keller, “Web services on demand: WSLA-driven automated management,” IBM systems Journal, vol. 43, no. 1, pp. 136–158, 2004.

[3] S. Ferretti, V. Ghini, F. Panzieri, M. Pellegrini, E. Turrini, “QoS-aware clouds,” In Proc. of CLOUD 2010, the IEEE 3rd International Conference on Cloud Computing, 2010.

[4] L.J. Jin, V. Machiraju, and A. Sahai, “Analysis on service level agreement of web services,” HP, pp. 1-13, 2002.

[5] W. Sun et al., “The role of XML in service level agreements management,” In Proc. of ICSSSM '05, the 2005 International Conference on Services Systems and Services Management, Vol. 2, pp. 1118 – 1120, 2005.

[6] A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, T. Kakata, J. Pruyne, J. Rofrano, S. Tuecke, and M. Xu, “Web services agreement specification (ws-agreement),” Technical report, Open Grid Forum, 2007. http://www.ogf.org/documents/GFD.107.pdf.

[7] D.D. Lamanna, J. Skene, and W. Emmerich, “Slang: a language for defining service level agreements,” In Proc. of FTDCS '03, The Ninth IEEE Workshop on Future Trends of Distributed Computing Systems, pp. 100–106, 2003.

[8] H. Kaminski and M., Perry, “SLA automated negotiation manager for computing services,” In Proc. of the 8th IEEE International Conference on E-Commerce Technology and the 3rd IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services (CEC-EEE '06), pp. 47, 2006.

[9] D. Shomron, “SLA management for SaaS,” 2009. http://saasperspective.blogspot.com/2009/11/sla-management-in-saas.html

[10] Gartner, “Gartner says cloud consumers need brokerages to unlock the potential of cloud services,” 2009. http://www.gartner.com/it/page.jsp?id=1064712

[11] S. Venticinque, R. Aversa, B. Di Martino, M. Rak, and D. Petcu, “A cloud agency for SLA negotiation and management,” Euro-Par 2010 Parallel Processing Workshops, Lecture Notes in Computer Science, Vol. 6586, pp. 587–594, 2011.

[12] SLA@SOI project, “Empowering the service economy with SLA-aware infrastructures,” http://www.sla-at-soi.eu/

[13] W. Theilmann, J. Happe, C. Kotsokalis, A. Edmonds, K. Kearney, J. Lambea, “A reference architecture for multi-level SLA management,” Journal of Internet Engineering, Vol. 4(1), pp. 289–298, 2010.

[14] A. J. Ferrer, F. Hernández, J. Tordsson, E. Elmroth, A. Ali-Eldin, C. Zsigri, R. Sirvent, J. Guitart, R. M. Badia, and K. Djemame, “OPTIMIS: a holistic approach to cloud service provisioning,” Future Generation Computer Systems, Vol. 28 (1), pp. 66–77, 2012.

[15] B. Moore, Q.H. Mahmoud, “A service broker and business model for saas applications,” In Proc. of AICCSA 2009, the IEEE/ACS International Conference on Computer Systems and Applications pp. 322 – 329, 2009.

[16] G. Di Modica, O. Tomarchio, and L. Vita, “Dynamic SLAs management in service oriented environments,” Journal of Systems and Software, Vol. 82 (5), 2009.

[17] G. C. Silaghi, L. D. Şerban, and C. M. Litan, “A time-constrained SLA negotiation strategy in competitive computational grids,” Future Generation Computer Systems, Vol. 28(8), pp. 1303–1315, 2012.

[18] M. Resinas, P. Fernández, and R. Corchuelo, “A bargaining-specific architecture for supporting automated service agreement negotiation systems,” Science of Computer Programming, Vol. 77 (1), pp. 4–28, 2012.

[19] P. Hasselmeyer, C. Qu, L. Schubert, B. Koller, and P. Wieder, “Towards autonomous brokered SLA negotiation,” In Exploiting the Knowledge Economy: Issues, Applications, Case Studies, IOS Press, Amsterdam, 2006.

[20] S. Wang, et al., “Cloud model for service selection,” In Proc. of the 2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 666-671, 2011.

[21] A. Li, et al., "Comparing public-cloud providers," Internet Computing, IEEE, Vol. 15, pp. 50-53, 2011.

[22] G. Lai, C. Li, K. Sycara, J. Giampapa, “Literature review on multi-attribute negotiations,” Technical Report CMU-RI-TR-04-66, Carnegie Mellon University, Robotics Institute, 2004.

[23] A. AuYoung, L. Grit, J. Wiener, and J. Wilkes, “Service contracts and aggregate utility functions,” In Proc. of the IEEE Symposium on High Performance Distributed Computing, pp. 119–131, 2006.