Melanie Siebenhaar, The An Bin Nguyen, Ulrich Lampe, Dieter Schuller, Ralf Steinmetz: *Concurrent Negotiations in Cloud-based Systems*. In: K. Vanmechelen, J. Altmann, and O.F. Rana (Eds.): Proceedings of the 8th International Workshop on Economics of Grids, Clouds, Systems, and Services (GECON2011), in conjunction with ICSOC 2011, vol. Lecture Notes in Computer Science, no. 7150, pp. 17-31, Springer-Verlag Berlin Heidelberg, 2012.

Concurrent Negotiations in Cloud-based Systems

Melanie Siebenhaar, The An Binh Nguyen, Ulrich Lampe,
Dieter Schuller, and Ralf Steinmetz

Multimedia Communications Lab (KOM)
Technische Universität Darmstadt, Germany
Corresponding Author: melanie.siebenhaar@KOM.tu-darmstadt.de

Utilizing cloud-based services, consumers gain a high level of flexibility, but they cannot obtain individual Quality of Service guarantees or request service compositions according to their specific business needs. Therefore, appropriate mechanisms for an automated negotiation of Quality of Service parameters are required that do not only consider the individual business objectives and strategies of the negotiation partners involved, but do also account for the dependencies between the different services and service tiers in cloud computing. This enables enterprises to increase the quality and flexibility of their business processes and lays the foundation for market-based complex service provisioning. In this paper, we present one such negotiation approach and evaluate the application of different negotiation strategies.

1 Introduction

In recent years, enterprises have experienced an increasing need to provide a flexible and competitive business process infrastructure, with IT systems serving as the key enabler. Several computing paradigms have been introduced, which promise to provide more IT flexibility. The latest of these is cloud computing, which enables on-demand access to arbitrary resources as a service (e.g., storage). Although cloud computing is already applied in practice, current vendors have only concentrated their effort on specific issues (e.g., scalability). To date, they offer no or only limited support for dynamic negotiation of individual Quality of Service (QoS) guarantees [3]. Hence, obtaining cloud services according to consumers' specific business constraints remains an open issue. Yet, quality parameters such as reliability and availability are crucial in a business environment. In order to retain control of the service quality, Service Level Agreements (SLAs) can be negotiated to ensure the desired quality is maintained. An SLA is a formal agreement, i.e., a contract between two parties and specifies the consumer's objectives (e.g., QoS parameters) that must be fulfilled by a provider and penalties in case of violations.

Negotiating individual SLAs in cloud computing is challenging. Negotiations typically involve consumers and providers with conflicting interests. Consumers usually want to obtain a high-quality service at low costs. Likewise, providers try to achieve the highest possible profit in line with demand, given their currently available QoS levels and capacities. Finally, consumers and providers each have to decide on a promising negotiation strategy. Since both parties try to achieve the highest possible utility and, due to the business context, do not want to disclose too much private information (e.g., business goals, cost factors), a negotiation of QoS parameters is necessary.

In addition, there can be multiple competing providers in the market, which offer the same type of service, but with different properties. Therefore, consumers wish to explore the heterogeneous service properties of different providers in advance in order to determine the most suitable services before establishing an agreement with a specific provider. Furthermore, if consumers want to combine services from different cloud providers, also the composition must satisfy the consumers' QoS requirements. Hence, it is necessary to negotiate concurrently with multiple providers from a consumer's point of view. Likewise, it is also required to conduct concurrent negotiations with multiple consumers from a provider's point of view in order to determine the consumers which generate the highest possible profit. Further challenges arise in relation to the several dependencies between the different service tiers (e.g., software, infrastructure) [2] in cloud computing. While service consumers wish to obtain specific services from different service providers, the service providers in turn must acquire the necessary resources for service execution from infrastructure providers. Hence, besides the issue of resource availability, also the adherence to SLAs across different administrative domains has to be considered.

Due to the large number of cloud providers and cloud consumers, the information exchange between the parties involved is very complex. Thus, a dynamic, scalable, and automated approach is required for negotiating SLAs with multiple providers across heterogeneous domains.

In the past, several approaches have been proposed for SLA negotiation in different fields of research (e.g., [17], [19]). However, very few effective solutions for automated negotiation have been provided so far [18]. In this paper, we present an approach for negotiating SLAs with multiple cloud providers across multiple tiers. We propose a cloud negotiation support system (CNSS) that can be employed on every service tier. Furthermore, we compare different negotiation strategies to be applied in our scenario.

The remainder of the paper is structured as follows. Section 2 discusses related approaches in the field of concurrent negotiations with multiple parties. In Section 3, the requirements for negotiating SLAs with multiple cloud providers across heterogeneous domains are described. Section 4 introduces our negotiation approach and Section 5 presents initial experimental results of our evaluation of different negotiation strategies. The paper closes with a conclusion and future directions in Section 6.

Table 1. Overview of Related Work

Publication	*Mult. Issues*	*Mult. Cust.\Prov.*	*Concurrency*	*Coordination*	*Protocol*	*Strategy*
This approach	×	××	×	×	Mod. Extended CNP	Time-dependent
Aknine et al. [1]	–	××	×	–	Extended CNP	–
Chhetri et al. [4]	×	–×	–	×	Alternate Offer + CNP	–
Di Nitto et al. [7]	×	××	–	×	–	Optimization
Sim and Shi [13]	–	–×	×	×	Alternate Offer	P_{Renege} + Time
Dang and Huhns [6]	×	××	×	–	Alternate Offer	–
Sim [15]	–	××	×	×	Alternate Offer	Market-driven

2 Related Work

Several approaches for concurrent negotiations with multiple providers have been proposed so far in different fields of research (cf. Table 1).

Aknine et al. [1] present an extension of the contract net protocol (CNP) [16] in order to support concurrent many-to-many negotiations. Basically, the contract net protocol is a simple one round-based protocol for task distribution in IT systems. The authors introduce a two-phase negotiation process which enables prospective contractors to overbid other offers. However, strategies and the assignment of multiple tasks are not considered in their approach.

Chhetri et al. [4] also adapt the CNP and propose a coordinated architecture for agent-based SLA negotiations. In their architecture, a global coordinator agent is responsible for determining a service composition according to consumer's QoS requirements. Local negotiation agents in turn conduct negotiations with multiple providers in order to achieve an agreement for a specific service type. A negotiation agent negotiates with multiple providers in an iterative manner over multiple rounds. No bidding strategies are specified.

In [7], Di Nitto et al. suggest a search-based solution for SLA negotiation. Similar to the work in [4], each negotiation participant is represented by a coordinator and several negotiators. In contrast to [4], Di Nitto et al. make use of an intermediate mediator in the form of a marketplace. The marketplace issues proposals to the participants based on an optimization algorithm in order to improve the convergence of the offers. The authors do not explicitly state a protocol for message exchange and private information is disclosed to the marketplace.

Sim and Shi [13] propose an approach for allocating multiple types of resources to perform a particular computation in grid computing. Consumers and providers apply a time-dependent strategy and are allowed to break an intermediate contract by paying a penalty fee. In addition, the strategy of the consumers is based on the calculation of the expected utility of the proposals and the probabilities that providers will renege from an intermediate contract. This approach requires the management of commitment and decommitment of contracts. Furthermore, a breach of contract may affect reputation.

In [6], Dang and Huhns adapt the alternate offers protocol in order to support concurrent negotiations with multiple issues in case of many-to-many negotiations. Their approach is based on the extended CNP [1] and also introduces two negotiation phases. Since counter-proposals can overbid formal proposals, it is obvious, that the negotiation may result in an infinite loop. The authors argue that this situation can be prevented by enforcing time constraints. However, time-dependent strategies are not considered in their approach.

Sim [15] focuses on market-based SLA negotiations in cloud computing. He considers a three-tier model where negotiation takes place between consumers, brokers and resource providers. A market-driven strategy is applied, where the concession amount depends on time, trading alternatives and competition. In his model, an agent can also renege from an intermediate contract by paying a penalty fee. The main goal for consumers is price minimization. However, for general SLA negotiation support, other QoS parameters must also be considered.

Our approach allows to combine services from multiple providers and to negotiate individual QoS parameters across multiple tiers. We introduce coordinating entities to manage the composition and make use of a two-phase protocol for concurrent negotiations with multiple providers. In the second phase of the protocol, we permit multiple overbidding. Furthermore, we apply time-dependent strategies, which stop the overbidding process if necessary.

3 Negotiation Model and Requirements

The work at hand focuses on service composition in cloud computing, where m services from different providers can be combined to form a complex service. Our approach is based on a market model where consumers submit their requirements to brokers in terms of desired functional and non-functional properties for specific services [3]. The brokers have access to a service registry in the market. By querying the registry, a broker is able to determine the sets of the most suitable cloud providers for the different m services based on the functional properties. A broker acting on behalf of a consumer conducts the negotiation of the non-functional properties, i.e., QoS parameters for the m services resulting in an agreement or in the breakdown in negotiations. For this purpose, a broker has to start m negotiation processes. Each process consists of concurrent one-to-many negotiations for a specific type of service with a set of providers. In addition, it may also be necessary for a service provider to initiate further negotiations with multiple providers on the lower resource level in order to lease infrastructure for the deployment of a specific service instance. To realize our approach, an appropriate negotiation mechanism is required to conduct concurrent negotiations with multiple providers, even across multiple tiers.

3.1 One-To-Many Negotiations

Basically, a negotiation mechanism consists of two components: a negotiation protocol and the negotiation strategies of the negotiating parties [9]. The negotiation protocol specifies the rules for interaction (i.e., message exchange, conditions for agreement) between the negotiating parties and the negotiation strategies must be compatible with the applied protocol. A negotiation strategy defines the sequence of actions planned to make during negotiation by a participant.

In our scenario, each negotiation concerns a specific service and the negotiating parties have multiple conflicting interests (e.g., price). The conflicting interests refer to the negotiable values in the form of non-functional parameters of a service. Similar to Microsoft Office[1] and research conducted in the area of Web services [19], we assume that each service is offered in the form of different priced packages (e.g., Gold, Silver, Bronze).

As input for negotiation, consumers and providers must specify their requirements for each service. We assume that the consumer specifies ranges for the several QoS parameters reflecting the lower and upper bounds he is willing to accept (e.g., response time between 5 ms and 10 ms). Since some parameters may be more important than others, we also assume that the consumers and providers specify weights for each QoS parameter. Furthermore, a goal is required on both, consumer and provider side, in order to make decisions during each round of the negotiation process and, ultimately, to reach an agreement. On both sides, the goal can be expressed based on the expected benefit from a given service offer. From a consumer's perspective, this can be mathematically expressed as follows: Given a set of m attributes $X = \{x_1, ..., x_m\}$ and different weights $W = \{w_1, ..., w_m\}$ for the attributes with $\sum_{i=1}^{m} w_i = 1$ for a desired cloud-based service. Let $U_e^t = f(W, X)$ be the expected utility of the service consumer in round t and let $U_e^{'}$ be the minimum expected utility of the service consumer. Further, let $O^t = \{o_1, ..., o_n\}$ be the set of service offers provided by n cloud providers in round t and $U^t(o_i)$ be the utility of the consumer for the service offer from the i^{th} provider. Given these parameters, the general goal is to choose a service offer o_i during negotiation that results in a minimum distance between the consumer's expected utility U_e^t and the utility $U^t(o_i)$ of the service offer o_i to the consumer (cf. Equation 1).

$$\arg\min_i f(i) = \left|(U_e^t - U^t(o_i))\right| \text{ where } U^t(o_i), U_e^t \geq U_e^{'} \tag{1}$$

3.2 Complex Cloud Service Negotiation Requirements

Several implicit and explicit assumptions are already part of the one-to-many negotiation model mentioned above. However, further requirements must be taken into account for our global cloud service composition scenario.

[1] http://office.microsoft.com/en-us/buy/office-2010-which-suite-is-right-for-you-FX101825640.aspx [last access: 2011-10-01]

- **Common Protocol:** In order to enable interoperability, consumers and providers must agree on a common protocol first before participating in a negotiation. This issue is addressed by conducting so-called meta-negotiations (e.g., [3]), which are performed before the actual negotiation takes place. Meta-negotiations are not part of our work.
- **Imperfect Information:** To allow for optimal negotiation results, the scoring information must be public [11]. However, if the parties have competing interests, the parties do not want to disclose their strategies to other parties. Hence, some information must be private (e.g., decision models) and other information must be public (e.g., expected QoS parameters of a consumer).
- **Time Limits:** We assume a given time limit as a condition for the completion of negotiations, since service brokers do not have an infinite amount of time to reach an agreement [12].
- **Administrative Domains:** In a common three-tier cloud model [2], users need to negotiate with service providers, who must, in turn, negotiate with resource providers to acquire the required resources. The QoS levels provided by service providers depend on the QoS levels provided by the resource providers. Hence, SLAs must be established across tiers while considering the dependencies between the QoS levels on different tiers.
- **Coordination:** Besides the dependencies across tiers, it is also necessary to coordinate the service composition. Coordinators must be established in order to balance the dependencies between the QoS parameters of the different services and to manage the available resources (e.g., [4], [7]).
- **Security:** The communication between the negotiating parties must be performed in a secure manner (e.g., SSL encryption of messages). These issues are not considered in our work.

4 Approach

Based on the assumptions and requirements outlined in the last section, we propose an approach for concurrent negotiations in cloud-based systems in order to combine services from multiple providers. From the discussion of related work in Section 2 it follows that coordinating entities are required when consumers want to combine services from multiple providers. In addition, other entities are necessary, which are controlled by coordinators, to conduct the negotiations.

4.1 Negotiation Architecture

Our negotiation architecture is based on the models proposed by Di Nitto et al. [7] and Chhetri et al. [4]. Each entity participating in the negotiation is represented in the form of a Cloud Negotiation Support System (CNSS) as depicted in Figure 1. After having received a service composition request from a consumer, a broker determines the most suitable cloud providers for each of the different services based on the consumer's functional requirements. Subsequently, the service composition request is passed to the CNSS of the broker together with the

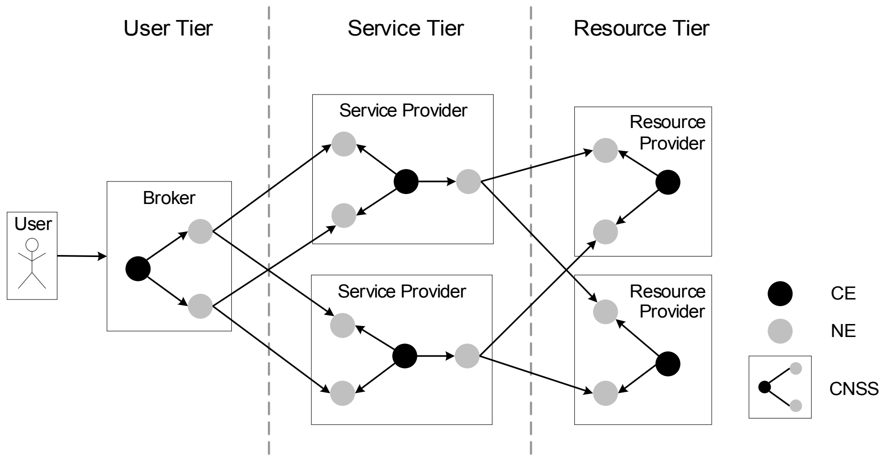


Fig. 1. Negotiation Architecture

provider lists. The CNSS then creates a coordinating entity (CE) responsible for managing the concurrent negotiations with the providers. The CE in turn creates several negotiating entities (NE) according to the number of services in the composition request. Each of them has the task to negotiate concurrently with multiple providers over the QoS parameters of a particular service. It can be observed in Figure 1, that this setup enables negotiations across different tiers. Service providers who want to obtain resources from resource providers initiate concurrent negotiations in a similar way. That is, a CE creates the required number of NEs which start negotiations with the respective resource providers. In order to account for the QoS parameters resulting from the negotiations with the brokers on consumer side, the CE of a service provider's CNSS must also observe these negotiations. Hence, the tasks of a CE can be defined from two perspectives.

- **CE on behalf of consumers:** Before initiating negotiations with providers, a CE has to split the consumer's requirements into global requirements for the composition and local requirements for each service that is part of the composition. The global requirements are observed by the CE and the local requirements are passed to the respective NEs. Since the CE manages the whole negotiation process, additional information such as session deadlines, reserve values or strategies is also passed to the NEs.
- **CE on behalf of providers:** A CE on provider side must be aware of the currently available service levels or resources. Furthermore, CEs on provider level must pass this information to NEs and observe their behaviour during negotiation on a higher level.

Table 2. List of Message Types

Message ($CE \leftrightarrow NE$)	**Meaning**
inform	CE passes required negotiation information to NE
notify	NE uses notify message to report negotiation status
Message ($NE \leftrightarrow NE$)	**Meaning**
call for proposal	Negotiation is initiated by requesting for proposals
propose	Negotiation is initiated with initiated proposal
pre-proposal	Agent makes pre-proposal in the warum-up phase
pre-accept	Agent temporarily accepts a proposal
pre-reject	Agent temporarily rejects other agent
definitive-accept	Agent is formally bound to an agreement
definitive-proposal	Agent makes final (formal) proposal
definitive-reject	Agent rejects other agent completely
withdraw	Agent leaves the negotiation

4.2 Two-Phase Negotiation Protocol

Communication between negotiating parties is essential in order to reach an agreement. Furthermore, the different parties must be aware of the rules of the negotiation they are participating in. Basically, the rules for interaction can be specified in the form of a negotiation protocol. In our negotiation architecture, a CE creates, informs, and also commands NEs if required. The NEs in turn communicate with multiple other NEs outside their own CNSS by exchanging messages. The message types, which are required in our concurrent negotiation scenario, are depicted in Table 2. The proposed message types are based on the FIPA specification [8] and the work by Aknine et al. [1]. As prerequisite for determining preferences for service offers and for conducting automated negotiations, the issues must be expressed in a common and formal way. Our expression is based on that proposed by Sierra et al. [11]. Each issue is characterized by a constraint interval in terms of a minimum and maximum value. Typically, either the minimum or the maximum value for an issue will be part of the initial proposal. The remaining upper or lower bound of the interval represents the reserve value. In our scenario, a proposal is defined as follows.

Definition 1 (Proposal) *A proposal P is a message sent from a participant p to an opponent o and contains a set S of n issues and their values proposed by p. Each issue x_i with $i = 1...n$ of the proposal corresponds to a specific quality parameter of a cloud service under negotiation. A proposal is valid, if every issue in the proposal is valid. An issue x_i is considered a valid issue, if its proposed value is within the intervals of p and o defined for this issue, i.e., $\forall x_i \in P$, $x_i \in [min_i^p, max_i^p]$ and $x_i \in [min_i^o, max_i^o]$.*

During negotiation, each party aims to maximize its utility. Since the negotiating parties have conflicting interests, concessions have to be made in order to reach a mutual agreement. The amount of concession depends on the applied strategy. In order to reach an agreement, the negotiating parties typically alternate in making proposals. Based on the issues in the proposal, the utility for a proposal is calculated. Intuitively, a participant prefers one proposal over another, if the utility of the opponent's proposal to the participant is equal or higher than the utility of the participant's own proposal. In this case, a proposal is acceptable. However, since the negotiations in our scenario involve multiple parties, more than a single acceptable proposal may exist. Therefore, a proposal can be (temporarily) accepted, if the following condition holds [5]:

Definition 2 (Acceptance Condition) *Let B_i with $i = 1...n$ be a set of providers a consumer C negotiates with. A Proposal $P_{B_i \to C}$ sent from provider B_i to consumer C can be accepted by C, if $U_C(P_{C \to B_i}) \leq U_C(P_{B_i \to C})$ and $U_C(P_{B_i \to C}) = max(U_C(P_{B_j \to C}))\, \forall P_{B_j \to C}$.*

After having given a formal expression for the negotiation issues and an acceptance condition for proposals, we can now elaborate on the different message types of the negotiation protocol applied in our scenario. The negotiation protocol is based on the protocols proposed by Dang and Huhns [6] and Aknine et al. [1]. As in their approaches, we also introduce two phases into the negotiation process: a *warm-up* and a *countdown* phase. During the warm-up phase, proposals are exchanged to find mutual interests. Afterwards, providers have to compete with other providers for the best proposal during the countdown phase. A two-phase approach has been chosen, since this protocol supports negotiations in a flexible manner. The negotiation participants are not immediately bound to the first accepted proposal. Instead, the negotiation continues, which allows the participants to find more appropriate proposals. The different message types are described in the following.

Message exchange between CE and NE: CEs create NEs to conduct concurrent negotiations with multiple providers for a particular type of service. An *inform* message is sent from a CE to an NE to provide the NE with all the information required for a negotiation (e.g., deadline, reserve value, strategy). In addition, a CE can send an inform message to interfere in the negotiation process (e.g., withdraw an NE from a negotiation). An NE in turn is able to report the negotiation status to a CE by sending a *notify* message.

Message exchange between NE and NE: NEs initiate negotiations with providers upon request of a CE by either sending an initial *proposal* or a *call for proposal* message to their opponents. In the latter case, NEs request initial proposals from providers. The negotiation is carried out in rounds. During the warm-up phase, the participants alternate in sending *pre-proposal* and counter pre-proposal messages until there is at least one acceptable proposal. An NE then determines the best proposal and sends a *pre-accept* message to the owner of this proposal and *pre-reject* messages to all other opponents. Subsequently, the receiver of the pre-accept message formulates a *definitive-proposal* message and sends it back to the NE. Other opponents, who received a pre-reject message, may still participate in the negotiation. They can formulate pre-proposal messages and try to overbid the currently best proposal. Even a definitive-proposal can be overbid by a pre-proposal sent by one of the participants in the last round. The overbidding process continues until a definitive-proposal is accepted as the best proposal. The owner of this proposal is notified by sending a *definitive-accept* message. All others receive a *definitive-reject* message and the negotiation is over. The last message type to be mentioned is the *withdraw* message. Such a message can be sent in any round during negotiation by a participant, who wants to leave the negotiation.

Simply by using the negotiation protocol, there is no guarantee that an agreement can be reached. Therefore, appropriate negotiation strategies are also necessary in order to increase the probability of reaching an agreement.

4.3 Negotiation Strategies

Negotiation strategies do not only affect the probability of reaching an agreement, but also affect the quality of an agreement (e.g., in terms of the achieved utility). Different strategies have been proposed in related work so far. Since time plays an important role in negotiation, even more so when concurrent negotiations are conducted between multiple competing parties, we apply and assess time-dependent strategies in our scenario.

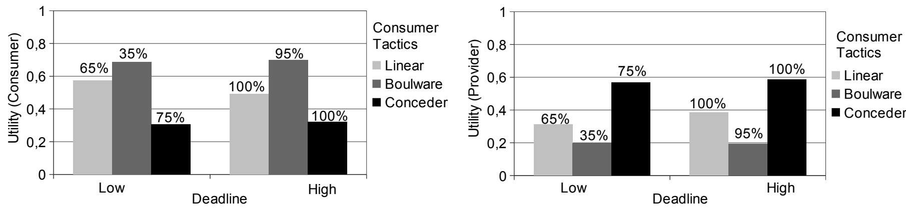
In Section 4.2 we stated that each participant defines ranges $x_i \in [min_i, max_i]$ for each negotiable issue. The utility values for an issue are typically defined as a value within a range from 0 to 1, with 1 representing the highest utility. Concerning the negotiation strategy, we find it intuitive to make an initial offer with the most preferred value for each issue and make concessions during negotiation to reach an agreement. Therefore, the utility value for the i-th issue x_i of a proposal P can be calculated by a utility function U as follows (e.g., [19], [14]):

$$U_P(x_i) = \begin{cases} \frac{max_i - x_i}{max_i - min_i} & \text{if a decrease of } x_i \text{ leads to a higher utility} \\ \frac{x_i - min_i}{max_i - min_i} & \text{if an increase of } x_i \text{ leads to a higher utility} \end{cases}$$

It can be obtained from the equations that the most preferred value for an issue is the minimum value in the first case and the maximum value in the second case. In order to determine the concession amounts for each issue in round t, we apply the time-dependent tactics as proposed by Sierra et al. [11]. Basically, a negotiation strategy can be defined by using one or more tactics. A pure strategy applies a single tactic to compute the next value for an issue, while a mixed strategy makes use of a weighted combination of tactics [10]. In our scenario, we use pure strategies based on the time-dependent tactic family. Using a time-dependent tactic, a counter proposal for issue i at round $t \leq t_{max}$ with t_{max} representing the deadline is calculated as follows [11]:

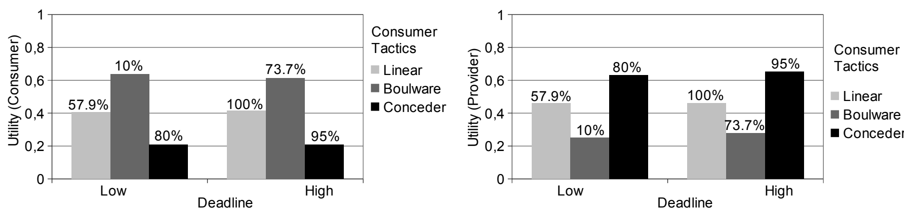
$$x_i = \begin{cases} min_i + \alpha_i(t)(max_i - min_i) & \text{if } U_P(x_i) \text{ is decreasing} \\ min_i + (1 - \alpha_i(t))(max_i - min_i) & \text{if } U_P(x_i) \text{ is increasing} \end{cases}$$

Function $\alpha_i(t)$ determines the concession amount made for issue i in round t. Besides the exponential function stated below, also a polynomial function can be used. We decided to use the exponential function, since it concedes slower at the beginning. The function is calculated as follows [11]:



(a) Linear Tactic for Provider (b) Linear Tactic for Provider

Fig. 2. Average Consumer and Provider Utility and Linear Tactic for Provider



(a) Boulware Tactic for Provider (b) Boulware Tactic for Provider

Fig. 3. Average Consumer and Provider Utility and Boulware Tactic for Provider

$$\alpha_i(t) = e^{(1-\frac{t}{t_{max}})^{\beta} \ln k_i}$$

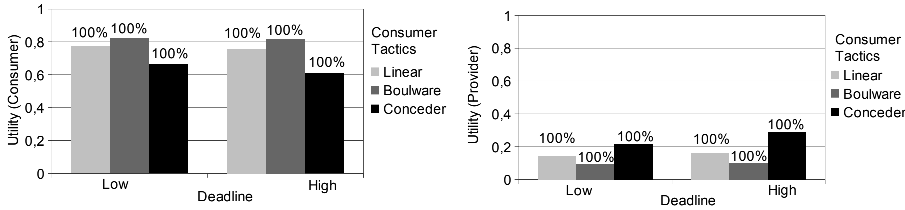
The initial proposal can be obtained from $\alpha_i(t)$ by multiplying k_i with the size of the constraint interval. The constant k_i can be chosen based on experience. The tactics are parameterized by the value β, which influences the concession amount [11]:

– Conceder ($\beta > 1$): the negotiating party makes a great concession already at the beginning and soon reaches the reserve value
– Linear ($\beta = 1$): the negotiating party makes an equal[2] concession in each round during negotiation
– Boulware ($\beta < 1$): the negotiating party maintains its proposed value nearly all the time and only makes a great concession shortly before the deadline

5 Experimental Results

This section presents initial experimental results of our approach. For this purpose, the two-phase negotiation protocol as described in Section 4.2 has been implemented. Furthermore, we perform a comparison of the different time-dependent

[2] when using the exponential $\alpha_i(t)$ function concession is made in near constant rate



(a) Conceder Tactic for Provider (b) Conceder Tactic for Provider

Fig. 4. Average Consumer and Provider Utility and Conceder Tactic for Provider

strategies applied for concurrent one-to-many negotiations in our cloud-based scenario. The evaluation is a proof-of-concept for the proposed approach and at the same time analyzes the influence of different provider and consumer strategies as well as different deadlines on the success rate and on consumer and provider utility. The simulations have been performed on a laptop with a Core 2 Duo 2.40GHz processor with 2 GB RAM and Ubuntu 11.04 as operating system. The different negotiating entities are represented by software agents. Repast Simphony[3] has been used as agent framework. Our setup for the experiments uses the settings applied in [10] as guideline. In each experiment, a consumer negotiates concurrently with 10 cloud providers and a specific type of strategy is applied one each side. All providers use the same type of strategy, but the strategic parameter β (i.e., the concession amount) can be different. The strategies on consumer and provider side do not change during negotiation. We performed separate experiments to analyze the effect of different strategies and deadlines on the utility values and success rate. Each strategy has been evaluated with respect to the other three time-dependent strategies using either low or high deadlines on both sides. Hence, comparing 3 consumer strategies with 3 provider strategies using 2 different deadlines we obtain a total number of 18 experiments. For each experiment, 20 test cases have been generated. The β value of the consumer is 0.5, 1, and 2.5 for strategy boulware, linear, and conceder, respectively. Providers' β values are randomly selected between (0.0,1.0) if boulware strategy is used, are equal to 1.0 if the strategy is linear, and are in range (1.0,5.0] if the conceder strategy is applied. In each experiment, consumer and providers either randomly select their deadline between [10,15] rounds in case of a low deadline experiment or between [25,30] rounds in case of a high deadline experiment. The initial proposals of consumer and providers are also generated at random for each of the 20 scenarios, but the 20 scenarios are fixed for all of the 18 experiments. Proposals in our experiments comprise three different issues: price, response time, and availability. The ranges of the different issues used for proposal generation are listed in Table 3.

[3] http://repast.sourceforge.net/

The outcomes of the evaluation are depicted in Figures 2 to 4. The Figures display the average utility either to consumer or provider of all successful deals depending on the strategies and deadlines used in each experiment. In addition, each bar indicates the percentage of successful deals (i.e., success rate). For each experiment, average utility and success rate are calculated as follows:

$$avg.\ utility_{cust/prov} = \frac{\sum utility_{cust/prov}}{|Deals_{succ}|} \qquad succ.\ rate = \frac{|Deals_{succ}|}{|Deals_{all}|} \tag{2}$$

The average run times of a single scenario in case of low and high deadlines are 50 ms and 80 ms, respectively. The simulation of a single experiment (20 scenarios) takes between 0.7 and 2.5 seconds. It can be obtained from the results that the deadline has only a marginal effect on the utility values of the consumer and provider, but even more on the success rate of the negotiations. Therefore, it can be more beneficial to make use of a higher deadline in order to reach an agreement. Nevertheless, in case of low deadlines, the success rates of the different strategies differ considerably. Hence, if there is an urgent need for a consumer to reach an agreement, applying a conceder strategy or, as the second best option, a linear strategy, gives a consumer the biggest chance to succeed. On average, the linear strategy provides a greater utility to the consumer than the conceder strategy. The boulware strategy is the dominant strategy on consumer side concerning the amount of utility, but only with little chance to succeed in case of low deadlines. On provider side, the boulware and the linear strategy provide the highest utility to the provider. They do not differ very much concerning the amount of utility, but their success rates slightly differ depending on the strategy applied by the consumer. The conceder strategy should not be applied from a provider's point of view, since it achieves the worst utility values in all experiments. Finally, the utility values of consumer and provider must be compared to each other. If the consumer applies a boulware strategy, the amount of utility to the consumer is always higher than the amount of utility to the provider. The reverse case occurs, when a consumer applies the conceder strategy, except for this situation when the provider also applies the conceder strategy. Nearly similar utility values are obtained on average, when the consumer applies a linear strategy and the provider applies a boulware strategy.

Summarizing, if we consider the best trade-off between amount of utility and success rate, a consumer should apply a linear strategy in case of low deadlines and a boulware strategy in case of high deadlines. On the provider side, the linear strategy has higher success rates on average, but the boulware strategy provides

Table 3. Ranges for Consumer and Provider Proposals

Issue	**Consumer**		**Provider**	
Price	min[8,12]	max[18,22]	min[10,15]	max[20,25]
Resp. Time	min[1,5]	max[10,15]	min[1,7]	max[13,17]
Availability	min[-99.999,95]	max[-85,-80]	min[-99.9,-90]	max[-80,-70]

a slightly higher amount of utility to the provider. If a balance must be achieved between the average utility values of consumer and provider, a consumer should apply a linear strategy while a provider uses a boulware strategy.

6 Conclusion

In this paper, we have presented an approach for concurrent negotiations in cloud-based systems. We have introduced a CNSS-based architecture and a two-phase negotiation protocol to conduct negotiations with multiple providers across multiple tiers. In addition, we have evaluated the applicability of three different time-dependent strategies for negotiation in our scenario. The results reveal that the deadline has a large effect on the success rate, but less on the achieved utility. Furthermore, we have seen in low deadline experiments that the best strategy of a consumer in terms of utility and success rate highly depends on the strategy of the opponent. Therefore, appropriate mechanisms have to be developed to determine a previously unknown strategy or new strategy variants of the opponent. This permits an agent acting on behalf of a consumer to better react to the providers' strategies. For this purpose, also appropriate decision support systems have to be developed, which choose the most promising strategies for consumers. Further enhancements of our approach are the evaluation of other negotiation strategies (e.g., resource dependent tactics) and the assessment of mixed strategies. Finally, a coordinator mechanism must be developed, which specifies the actions of a CE, when and how to interfere in negotiations.

References

1. Aknine, S., Pinson, S., Shakun, M.: An Extended Multi-Agent Negotiation Protocol. Autonomous Agents and Multi-Agent Systems 8(1), 5–45 (2004)
2. Armbrust, M., Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., et al.: A View of Cloud Computing. Communications of the ACM 53(4), 50–58 (2010)
3. Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., Brandic, I.: Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility. Future Generation Computer Systems 25(6), 599–616 (2009)
4. Chhetri, M.B., Lin, J., Goh, S., Zhang, J.Y., Kowalczyk, R., Yan, J.: A Coordinated Architecture for the Agent-based Service Level Agreement Negotiation of Web Service Composition. In: Proceedings of the Australian Software Engineering Conference. pp. 90–99 (2006)
5. Dang, J., Huhns, M.: An Extended Protocol for Multiple-Issue Concurrent Negotiation. In: Proceedings of the 20th National Conference on Artificial Intelligence. pp. 65–70 (2005)
6. Dang, J., Huhns, M.: Concurrent Multiple-Issue Negotiation for Internet-based Services. IEEE Internet Computing 10(6), 42–49 (2006)
7. Di Nitto, E., Di Penta, M., Gambi, A., Ripa, G., Villani, M.: Negotiation of Service Level Agreements: An Architecture and a Search-based Approach. In: Proceedings of the 5th International Conference on Service-Oriented Computing (ICSOC 2007). pp. 295–306 (2010)

8. FIPA: Library Specification. http://www.fipa.org/specs/fipa00037/SC00037J.html
9. Lomuscio, A.R., Wooldridge, M., Jennings, N.R.: A Classification Scheme for Negotiation in Electronic Commerce. Agent-Mediated Electronic Commerce: A European Agent Link Perspective 12(1), 31–56 (2003)
10. Patel, D., Gupta, A.: An Adaptive Negotiation Strategy for Electronic Transactions. In: Proceedings of the 10th IEEE International Enterprise Distributed Object Computing Conference. pp. 243–252 (2006)
11. Sierra, C., Faratin, P., Jennings, N.R.: A Service-Oriented Negotiation Model between Autonomous Agents. In: Proceedings of the 8th European Workshop on Modelling Autonomous Agents in a Multi-Agent World: Multi-Agent Rationality. pp. 17–35 (1997)
12. Silaghi, G.C., Dan Şerban, L., Litan, C.M.: A Framework for Building Intelligent SLA Negotiation Strategies under Time Constraints. In: Proceedings of the 7th International Workshop on Economics of Grids, Clouds, Systems, and Services (GECON'10). pp. 48–61 (2010)
13. Sim, K., Shi, B.: Concurrent Negotiation and Coordination for Grid Resource Coallocation. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 40(3), 753–766 (2010)
14. Sim, K., Wang, S.: Flexible Negotiation Agent with Relaxed Decision Rules. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 34(3), 1602–1608 (2004)
15. Sim, K.M.: Towards Complex Negotiation for Cloud Economy. In: Proceedings of the 5th International Conference on Advances in Grid and Pervasive Computing (GPC 2010). pp. 395–406 (2010)
16. Smith, R.G.: The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. IEEE Transactions on Computers 29(12), 1104–1113 (1980)
17. Stantchev, V., Schröpfer, C.: Negotiating and Enforcing QoS and SLAs in Grid and Cloud Computing. In: Proceedings of the 4th International Conference on Advances in Grid and Pervasive Computing. pp. 25–35 (2009)
18. Wu, L., Buyya, R.: Service Level Agreement (SLA) in Utility Computing Systems. Tech. Rep. CLOUDS-TR-2010-5, Cloud Computing and Distributed Systems Laboratory, The University of Melbourne, Australia (September 2010)
19. Zulkernine, F., Martin, P., Craddock, C., Wilson, K.: A Policy-Based Middleware for Web Services SLA Negotiation. In: Proceedings of the 7th International Conference on Web Services. pp. 1043–1050 (2009)