Cloud Service Negotiation: Concession vs. Tradeoff Approaches

Xianrong Zheng, Patrick Martin
School of Computing
Queen's University
Kingston, Ontario, Canada K7L 3N6
{xianrong, martin}@cs.queensu.ca

Kathryn Brohman
School of Business
Queen's University
Kingston, Ontario, Canada K7L 3N6
kbrohman@business.queensu.ca

***Abstract*—For Cloud services, their non-functional properties like availability, reliability and security are important differentiators. However, service consumers and service providers may conflict over non-functional properties. In fact, the conflicts can be resolved via automated negotiation, which is considered as the most flexible approach to procure products and services. In this paper, we propose tradeoff approaches for Cloud service negotiation, and compare them with concession ones. As opposed to concession ones, tradeoff approaches do not reduce one's utility, but still can create a proposal attractive to its opponent. Indeed, simulation results show that tradeoff approaches outperform concession ones in terms of both individual utility and social benefit. However, simulation results also demonstrate that tradeoff approaches underperform concession ones in terms of success rate.**

Keywords-Cloud service negotiation; tradeoff approaches; concession approaches

I. INTRODUCTION

In Cloud computing, hardware, software and applications are all delivered as services over the Internet [2]. Cloud services refer to Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) [1]. Cloud services can be described with their functional and non-functional properties. Functional properties detail what is done, while non-functional properties detail how well it is done. For instance, the storage access offered by Amazon S3 is its functional properties, while reliability and storage price are two non-functional properties.

As Cloud services are delivered over the Internet, their non-functional properties like availability, reliability and security matter. However, service consumers and service providers may have conflicts over non-functional properties. For instance, service consumers may prefer a more reliable but less expensive service, whereas service providers may prefer to offer the opposite. How to resolve the conflicts over non-functional properties thus becomes crucial. In fact, the conflicts can be resolved via automated negotiation, which is considered as the most flexible approach to procure products and services.

Indeed, QoS measurement, QoS negotiation and QoS enforcement are three steps that comprise a whole process. Before we conduct negotiation, we need to specify a quality model that is objective, computable and verifiable. For instance, if we define reliability as the probability that there is no failure over n operations [13], it can be measured by

$$R(n) = 1 - \frac{f_n}{n} \qquad (1)$$

where f_n is the number of failures that occur over the n operations. Once a well-defined quality model is available, service consumers and service providers can negotiate QoS and then establish a SLA. For instance, a reliability of 90% could be specified in a Service Level Agreement (SLA). After that, service providers need to ensure their promised QoS with certain techniques. For instance, in order to ensure a reliability no less than 90%, a service provider whose storage devices are 70% reliable should replicate its data two times, which actually ensures a reliability of $1-(1-0.70)^2 = 91\%$. It is good enough!

This paper focuses on QoS negotiation. In terms of the art of negotiation, two negotiation strategies, which are decision making techniques, can be adopted: concession and tradeoff. With concession strategies, one reduces its utility gradually until no conflicts exist. As concession strategies reduce one's utility, they are considered when no alternatives exist. If preference gaps exist between two parties, it is possible to resolve conflicts without concession. With tradeoff strategies, instead, one yields on its less important attribute, but demands more on its more important attribute. As a result, a proposal attractive to one's opponent is created, but no utility is reduced.

The contributions of this paper are twofold. First, we propose tradeoff approaches for Cloud service negotiation. Second, we compare tradeoff approaches with concession ones, and analyze their pros and cons. The rest of the paper is structured as follows. Section II discusses related work. Section III describes a motivating example. Section IV proposes tradeoff approaches to procure Cloud services. Section V compares tradeoff approaches with concession ones using simulation. Section VI concludes the paper and discusses some future work.

II. RELATED WORK

Automated negotiation has been studied in multi-agent systems and electronic commerce for years, where a group of agents attempt to reach a mutually acceptable agreement on some matter [6, 7]. A few papers currently deal with Cloud service negotiation. We discuss them below.

A middleware architecture is proposed to enable dynamic configuration, management and optimization of Cloud services [9]. It is combined with a load balancer in order to respond effectively to QoS requirements of applications

978-0-7695-4691-9/12 $26.00 © 2012 IEEE
DOI 10.1109/CCGrid.2012.94

hosted in Clouds. However, no negotiation occurs here. In order to verify contract compliance, Service Level Checking (SLC) is applied to Clouds. Also, a mediation approach is employed to separate concerns about monitoring information collection and concerns about contract compliance verification [10]. Again, no negotiation happens.

Li and Jeng [11] state that automated negotiation is needed for Cloud services in order to reduce the cost and time of doing business. They present a negotiation model, however, no negotiation strategies are specified. In order to build adaptable Cloud services, automated negotiation is employed by Brandic et al., where service consumers can negotiate with service providers over non-functional requirements [4]. Also, document models are adopted for the specification of meta-negotiations and SLA mappings. Again, no negotiation strategies are defined. In order to negotiate Cloud services, concession strategies that consider time, outside options and market competition are proposed by Sim [3]. However, as concession strategies reduce one's utility, they are not the first choice in negotiation. In fact, they are considered only when something unfavorable happens, for instance, the deadline approaches.

III. A MOTIVATING EXAMPLE

A motivating example is presented here to illustrate that service consumers may conflict with service providers over non-functional properties. It is built on Amazon S3, and recapitulated in Table 1.

Assume, for simplicity, that two attributes are involved, i.e., reliability (measured in percentage per month) and storage price (measured in dollar per gigabyte and month). A service consumer (Alice) wants to have a reliability of 95%, but if not possible, a reliability of 80% is acceptable too. In other words, her preferred and reserved values for reliability are 95% and 80%, respectively. As her data stored on Amazon S3 is important, it is reasonable for Alice to prefer a high reliability. In contrast, the service provider (Amazon) has the ability to provide a reliability of 90%, but it only wants to offer a reliability of 75%. In other words, its preferred and reserved values for reliability are 75% and 90%, respectively. As it has to incur more costs and efforts to guarantee a high reliability, it also makes sense for Amazon to offer a commercially reasonable reliability. As it can be seen, a conflict over preferred reliability exists between Alice and Amazon (95% vs. 75%).

For her range of acceptable reliability, Alice's preferred and reserved values of storage price are 0.05 and 0.08, respectively. By contrast, for its range of acceptable reliability, Amazon's preferred and reserved values of storage price are 0.09 and 0.06, respectively. Here, storage price is a higher-is-better attribute for Amazon, whereas it is a lower-is-better one for Alice. As it can be seen, another conflict over preferred storage price exists between Alice and Amazon (0.05 vs. 0.09).

Alice and Amazon differ in their preferences for reliability and storage price. For Alice, she values reliability more than storage price, since she cannot afford the loss of her data. Hence, her weights for reliability and storage price are set as 0.9 and 0.1, respectively. By contrast, Amazon values storage price more than reliability, since it aims to make profits. Hence, its weights for reliability and storage price are set as 0.1 and 0.9, respectively.

Assumption 1 (*Incomplete Information*). Assume that Alice and Amazon are self-interested, and that they expose their partial preferences, but keep important parameters secret. In other words, we assume:

- Alice knows that Amazon cares more about storage price than reliability,
- Amazon realizes that Alice cares more about reliability than storage price.
- Both parties do not disclose their weights, and keep secret their preferred and reserved values of reliability and storage price.

In fact, the two conflicts described above cannot be resolved if no negotiation involves, while incomplete information makes the problem even harder. As preference gaps exist between Alice and Amazon, it is possible to resolve the conflicts without concession. In the following section, we describe tradeoff approaches in detail, and we compare them with concession ones in Section V.

IV. A DESCRIPTION OF TRADEOFF APPROACHES

When it comes to automated negotiation, negotiation protocols and negotiation strategies merit special attention [5, 6, 7]. Indeed, *negotiation protocols* specify the "rules of encounter" between negotiation agents [7]. In this paper, an alternating-offer protocol [12] is used for one-to-one negotiation. Once negotiation protocols are decided, *negotiation strategies*, which are decision making techniques that negotiation agents adopt to maximize their payoffs, become critical. As mentioned above, since preference gaps exist in the motivating example, it is possible to resolve the conflicts without concession. In fact, with tradeoff approaches, two parties can enlarge a pie first, and then divide it between them. It becomes a win-win game. Below, we first introduce utility functions, and then give an economic and an algorithmic description of tradeoff approaches.

TABLE I. CONFLICTS OVER RELIABITY AND STORAGE PRICE BETWEEN ALICE AND AMAZON

	Min.	Max.	Alice			Amazon		
			Reserved Value	Preferred Value	Weight	Reserved Value	Preferred Value	Weight
Reliability	0%	100%	80%	95%	0.9	90%	75%	0.1
Storage Price	0	0.1	0.08	0.05	0.1	0.06	0.09	0.9

A. Utility Functions

In economics, a utility function measures "the level of satisfaction a consumer receives from any basket of goods and services" [8]. It is adopted here to measure the level of satisfaction that users receive from Cloud services. For an attribute whose utility changes linearly with its value, a linear function can be used to calculate its utility. Let x be such an attribute, x_{best} its best value, and x_{worst} its worst value. Its utility, $u_1(x)$, can be calculated by

$$u_1(x) = \frac{x - x_{worst}}{x_{best} - x_{worst}} \quad (2)$$

where $0 \le u_1(x) \le 1$, $u_1(x_{best}) = 1$ and $u_1(x_{worst}) = 0$. It should be noted that x_{best} and x_{worst} are defined by negotiation agents on their own account. In particular, when x is a higher-is-better attribute, $x_{best} > x_{worst}$, and when x is a lower-is-better one, $x_{best} < x_{worst}$.

After that, a weighted sum function can be used to calculate the total utility of a proposal, p, containing n attributes that are additive $(n \in \mathbb{Z}, n \ge 2)$ by

$$u_2(p) = \sum_{i=1}^{n} w_i \cdot u_1(x_i) \quad (3)$$

where w_i is the weight of attribute x_i $(i = 1,...,n)$ and $\sum_{i=1}^{n} w_i = 1$.

Assumption 2 (*Linear Additive Utility Function*). Assume that the utilities of attributes reliability and storage price change linearly with their values, and that they are additive. In other words, we assume:

- Equation 2 can be used to calculate the utilities of reliability and storage price.
- Equation 3 can be used to calculate the total utility of a proposal that contains the two attributes.

In the motivating example, storage price is a higher-is-better attribute for Amazon, while reliability is a lower-is-better one. In fact, its best value of storage price is 0.1 and its worst value of price is 0. However, its best value of reliability is 0% and its worst value of reliability is 100%. As a result, the utility of its preferred value of storage price, i.e., 0.09, is calculated as (0.09 – 0) / (0.1 – 0) = 0.9. In a similar way, the utility of its preferred value of reliability, i.e., 75%, is calculated as (75% – 100%) / (0% – 100%) = 0.25. For Amazon, therefore, the total utility of its preferred offer is calculated as $0.9 \times 0.9 + 0.1 \times 0.25 = 0.835$.

It should be mentioned that the value, x', of an attribute and its corresponding utility, u', can be easily mapped to each other, since $u_1(x)$ is monotonic. Also, since $u_2(x)$ is additive, it is straightforward to extend the motivating example to handle more than two attributes, i.e., from two-dimensional to multi-dimensional.

B. An Economic Description

Tradeoff approaches work as follows. In preparing a counter proposal, the total utility of one party's reference proposal remains the same, but the values of some of its attributes are adjusted in favour of its opponent. If, for this reason, the utility of one attribute is decreased by a certain amount and that of another increased by the same amount, the total utility may not change. Even so, tradeoff approaches can encourage the opponent to accept the counter proposal, since its utility is increased in this case. Here, a *reference proposal* is built on one party's preferred proposal. In fact, a proposal received from the other party can be set as the reference proposal if the received utility is higher than its preferred one. It should be pointed out that some differences could happen if the second approach is adopted.

A graphical representation of tradeoff approaches is depicted in Fig. 1. Without loss of generality, a two-dimensional negotiation space, *xy*, is assumed here. Also, utility functions are assumed to be linear additive (i.e., Assumption 2). Let l_1 and l_2 be the indifference curves of one party's reference proposal and the preferred one of its opponent, respectively. In terms of economics, an *indifference curve* connects "a set of consumption baskets that yield the same level of satisfaction to the consumer" [8]. Again, let point A correspond to one party's initial proposal, and point B its counter proposal. Indeed, when one party makes a tradeoff, it moves along l_1 from point A to point B, which has the same utility but is closer to l_2, since $|BB'| < |AA'|$. In other words, it moves toward the preferences of its opponent, but no utility is reduced from its preferred proposal by doing so.

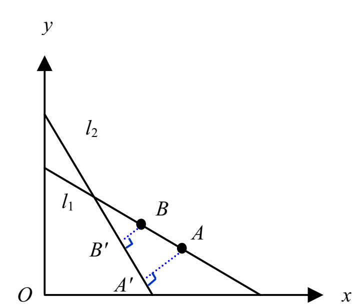

Fig. 1. An Illustration of Tradeoff Approaches

C. An Algorithmic Description

Tradeoff approaches are implemented in Algorithm 1, which is a distributed algorithm that involves two attributes. Here, one party's knowledge (i.e., Assumption 1) about the preferences of the other party is applied.

Algorithm 1 works as follows. First of all, in line 1, agent i sends V as a proposal to agent j and waits for a response. If agent j does not accept V and its counter proposal is not acceptable to agent i, tradeoff is used by agent i in the *while* loop of lines 2-23 to create a new proposal; otherwise, *true* is returned in line 24. Here, the acceptance criterion is that the utility received is no less than that of their respective reserved proposal. Next, in line 4, k is increased by one each

Algorithm 1: TRADEOFF (*V, W, F,* λ)

```
Input:  array V with raw values of two attributes;
        array W with weights of two attributes;
        array F with flags of two attributes. A flag here
        indicates whether an attribute is higher-is-better;
        parameter λ that indicates the degree of tradeoff at a
        time ( 0 < λ ≤ 1 )
Output: true if succeed and false otherwise
1   agent i sends V to agent j and waits for a response
2   while (agent j does not accept V and its counter proposal
3         is not acceptable to agent i)
4         k ← k + 1
5         u0 ← calculateUtility (V [0], W [0], F [0])
6         u1 ← calculateUtility (V [1], W [1], F [1])
7         usum ← aggreateUtility (V, W, F)
8         if W [0] < W [1] then
9              u0 ← u0 − (k × λ) × u0
10             u1 ← usum − u0
11        else
12             u1 ← u1 − (k × λ) × u1
13             u0 ← usum − u1
14        if u0 < 0 or  u1 < 0 then
15             return FALSE
16        else
17             V [0] ← restoreValue (u0, W [0], F [0])
18             V [1] ← restoreValue (u1, W [1], F [1])
19        if (V is out of bounds) then
20             return FALSE
21        else
22             agent i sends V to agent j and waits for a response
23  end
24  return TRUE
```

time the *while* loop repeats. In lines 5-6, function *calculateUtility* calculates the utilities u_0 and u_1 of agent *i*'s values *V [0]* and *V [1]*, respectively. In line 7, function *aggreateUtility* calculates the total utility u_{sum} of agent *i*'s preferred proposal *V*. In fact, the two functions implement the linear function and the weighted sum function in Equations 2 and 3, respectively. After that, in lines 8-13, if weights *W [0] < W [1]*, u_0 is reduced by $(k\times\lambda)\times u_0$ and u_1 thus becomes $u_{sum} - u_0$; otherwise, u_1 is reduced by $(k\times\lambda)\times u_1$ and u_0 becomes $u_{sum} - u_1$. In both cases, it ensures that agent *i*'s important attribute creates more utility. Finally, in lines 14-18, if $u_0 < 0$ or $u_1 < 0$, no values for *V [0]* or *V [1]* can be found, then tradeoff fails; otherwise, the new values can be found by function *restoreValue*, which is the inverse function of Equation 2. In lines 19-22, if *V* is out of bounds, *false* is returned; otherwise, agent *i* sends *V* whose values are adjusted but its utility remains the same for itself to agent *j* as a new proposal, and waits for a response. The process repeats until success or failure occurs. It should be mentioned that since u_0 (or u_1) is reduced by $\lambda\times u_0$ (or $\lambda\times u_1$) each time the *while* loop repeats, Algorithm 1 converges and terminates within $\lceil 1/\lambda \rceil$ rounds, where $0 < \lambda \le 1$. In other words, its running time can be bounded as $O(\lceil 1/\lambda \rceil)$.

The philosophy of tradeoff approaches can be summarized as follows. In order to create a counter proposal attractive to its opponent, one yields on its less important attribute, while it demands more on its more important attribute. If its opponent has opposite preferences, the less important attribute surprisingly creates more utility for its opponent. As a result, one eventually moves toward the preferences of its opponent.

V. Evaluation and Analysis

In this section, tradeoff approaches for Cloud service negotiation are evaluated using extensive simulations. All simulations are conducted on a Lenovo® ThinkCentre™ desktops with a 2.80 GHz Intel® Pentium™ Dual-Core CPU and a 2.96 GB RAM, running Microsoft® Window 7 Professional™ Operating System. The simulations are implemented with Java under NetBeans IDE 6.9.1 with JDK 6u21.

Algorithm 2: CONCESSION (*V, W, F,* λ)

```
Input:  array V with raw values of two attributes;
        array W with weights of two attributes;
        array F with flags of two attributes. A flag here
        indicates whether an attribute is higher-is-better;
        parameter λ that indicates the degree of concession at
        a time ( 0 < λ ≤ 1 )
Output: true if succeed and false otherwise
1   agent i sends V to agent j and waits for a response
2   while (agent j does not accept V and its counter proposal
3         is not acceptable to agent i)
4         k ← k + 1
5         u0 ← calculateUtility (V [0], W [0], F [0])
6         u1 ← calculateUtility (V [1], W [1], F [1])
7         u0 ← u0 − (k × λ) × u0
8         u1 ← u1 − (k × λ) × u1
9         if u0 < 0 or u1 < 0 then
10             return FALSE
11        else
12             V [0] ← restoreValue (u0, W [0], F [0])
13             V [1] ← restoreValue (u1, W [1], F [1])
14        if (V is out of bounds) then
15             return FALSE
16        else
17             agent i sends V to agent j and waits for a response
18  end
19  return TRUE
```

In order to make a comparison with concession approaches, we also implement them in Algorithm 2, which differs with Algorithm 1 in lines 7-13. Here, each time the *while* loop repeats, u_0 is reduced by $\lambda\times u_0$ while u_1 by $\lambda\times u_1$ in lines 7-8. Also, the acceptance criterion is that the utility received is no less than that of one's reserved proposal. However, when one serves as the offerer and makes a concession, its utility is allowed to be less than its reserved utility. Algorithm 2 terminates within $\lceil 1/\lambda \rceil$ rounds too, and so its time complexity is $O(\lceil 1/\lambda \rceil)$, where $0 < \lambda \le 1$. Below, we first describe negotiation scenarios and the negotiation process. After that, we report and analyze experimental results.

A. Negotiation Scenarios

Two sets of negotiation scenarios based on the motivating example are formed to simulate real negotiation environments, where the minimum and maximum values of reliability and storage price are the same as in Table I. The first set of negotiation scenarios are shown in Table II, where the motivating example is relabeled as *S1*. Here, the gaps of the preferred values for reliability and storage price are set as 20% and 0.04, respectively. As a result, no agreement can be achieved without negotiation. If the conflicts are handled appropriately, however, it is possible that a settlement can be reached between two parties.

TABLE II. NEGOTIATION SCENARIOS *S1-S10*

		SP's Reserved Value	SP's Preferred Value	SP's Weight	SC's Reserved Value	SC's Preferred Value	SC's Weight
S1	**Rel.**	90%	75%	0.1	80%	95%	0.9
	Pri.	0.06	0.09	0.9	0.08	0.05	0.1
S2	**Rel.**	85%	70%	0.1	75%	90%	0.9
	Pri.	0.05	0.08	0.9	0.07	0.04	0.1
S3	**Rel.**	80%	65%	0.1	70%	85%	0.9
	Pri.	0.04	0.07	0.9	0.06	0.03	0.1
S4	**Rel.**	75%	60%	0.1	65%	80%	0.9
	Pri.	0.03	0.06	0.9	0.05	0.02	0.1
S5	**Rel.**	70%	55%	0.1	60%	75%	0.9
	Pri.	0.02	0.05	0.9	0.04	0.01	0.1
S6	**Rel.**	90%	75%	0.1	80%	95%	0.9
	Pri.	0.02	0.05	0.9	0.04	0.01	0.1
S7	**Rel.**	85%	70%	0.1	75%	90%	0.9
	Pri.	0.03	0.06	0.9	0.05	0.02	0.1
S8	**Rel.**	80%	65%	0.1	70%	85%	0.9
	Pri.	0.04	0.07	0.9	0.06	0.03	0.1
S9	**Rel.**	75%	60%	0.1	65%	80%	0.9
	Pri.	0.05	0.08	0.9	0.07	0.04	0.1
S10	**Rel.**	70%	55%	0.1	60%	75%	0.9
	Pri.	0.06	0.09	0.9	0.08	0.05	0.1

TABLE III. NEGOTIATION SCENARIOS *S11-S20*

		SP's Reserved Value	SP's Preferred Value	SP's Weight	SC's Reserved Value	SC's Preferred Value	SC's Weight
S11	**Rel.**	90%	80%	0.1	85%	95%	0.9
	Pri.	0.07	0.09	0.9	0.08	0.06	0.1
S12	**Rel.**	85%	75%	0.1	80%	90%	0.9
	Pri.	0.06	0.08	0.9	0.07	0.05	0.1
S13	**Rel.**	80%	70%	0.1	75%	85%	0.9
	Pri.	0.05	0.07	0.9	0.06	0.04	0.1
S14	**Rel.**	75%	65%	0.1	70%	80%	0.9
	Pri.	0.04	0.06	0.9	0.05	0.03	0.1
S15	**Rel.**	70%	60%	0.1	65%	75%	0.9
	Pri.	0.03	0.05	0.9	0.04	0.02	0.1
S16	**Rel.**	90%	80%	0.1	85%	95%	0.9
	Pri.	0.03	0.05	0.9	0.04	0.02	0.1
S17	**Rel.**	85%	75%	0.1	80%	90%	0.9
	Pri.	0.04	0.06	0.9	0.05	0.03	0.1
S18	**Rel.**	80%	70%	0.1	75%	85%	0.9
	Pri.	0.05	0.07	0.9	0.06	0.04	0.1
S19	**Rel.**	75%	65%	0.1	70%	80%	0.9
	Pri.	0.06	0.08	0.9	0.07	0.05	0.1
S20	**Rel.**	70%	60%	0.1	85%	95%	0.9
	Pri.	0.07	0.09	0.9	0.04	0.02	0.1

The second set of negotiation scenarios are shown in Table III. Here, the gaps of the preferred values for reliability and storage price in *S11-S19* are set as 15% and 0.03, respectively. As for *S20*, the acceptable intervals for reliability and storage price do not overlap, so competitions are intensive. Hence, it is usually hard to resolve such conflicts.

B. Negotiation Process

An alternating-offer protocol is employed as the negotiation protocol, while tradeoff or concession is adopted as the negotiation strategies. The negotiation process works as follows. Firstly, a Service Provider (SP) sends its preferred proposal to a Service Consumer (SC). Then, if it is acceptable to SC, negotiation ends with an agreement established; otherwise, SC adopts tradeoff (or concession) approaches described in Algorithm 1 (or Algorithm 2) to create a counter proposal. After that, it sends the counter proposal to SP, and the negotiation process repeats. The process ends once a proposal or counter proposal is accepted, and it fails if no proposals or counter proposals make both parties satisfied. In all simulations, λ is set as 0.1 in both Algorithms 1 and 2.

C. Negotiaiton Results and Analysis

For negotiation scenarios in Table II, their simulation results are reported in Table IV. It can be seen that both concession and tradeoff approaches succeed in all ten negotiation scenarios. As an example, let us check *S1*, i.e., the motivating example. Here, SP's and SC's reserved utilities are 0.550 and 0.740, respectively, while their preferred utilities are 0.835 and 0.905, respectively. Under concession approaches, SC accepts the proposal offered by SP at round *six*, with reliability 80.0% and storage price 0.072. Accordingly, their received utilities become 0.668 and 0.748, respectively. It can be verified that their received utilities are greater than their reserved utilities (i.e., 0.668>0.550 and 0.748>0.740), respectively. Under tradeoff approaches, SC accepts the proposal offered by SP at round *eight*, with reliability 82.5% and storage price 0.091. Accordingly, their received utilities become 0.835 and 0.752, respectively. It can also be verified that their received utilities are greater than their reserved utilities (i.e., 0.835>0.550 and 0.752>0.740), respectively. Two observations are mentioned below.

Observation 1 (*Individual Utility*). Tradeoff approaches produce more utility than concession approaches for one who can offer an attractive proposal to its opponent.

As an example, let us check *S2*. When both parties switch from concession approaches to tradeoff ones, SC's received utility is increased from 0.720 to 0.870, while SP's received utility is decreased from 0.600 to 0.477. Here, SC serves as a *successful offerer,* whose proposal is accepted by its opponent in the last round.

TABLE IV. EXPERIMENTAL RESULTS FOR *S1-S10*

	Concession		**Tradeoff**	
	Offer	**Utilities**	**Offer**	**Utilities**
S1	SP→SC: Succeed at Round 6 with Rel 80.0% & Pri 0.072	SP: 0.668, SC: 0.748 Total: 1.416	SP → SC: Succeed at Round 8 with Rel 82.5% & Pri 0.091	SP: 0.835, SC: 0.752 Total: 1.587
S2	SP → SC: Succeed at Round 6 with Rel 76.0% & Pri 0.064	SP: 0.600, SC: 0.720 Total: 1.320	SC→ SP: Succeed at Round 7 with Rel 91.3% & Pri 0.052	SP: 0.477, SC: 0.870 Total: 1.347
S3	SP → SC: Succeed at Round 6 with Rel 72.0% & Pri 0.056	SP: 0.532, SC: 0.692 Total: 1.224	SP → SC: Succeed at Round 6 with Rel 72.0% & Pri 0.071	SP: 0.665, SC: 0.677 Total: 1.342
S4	SP → SC: Succeed at Round 6 with Rel 68.0% & Pri 0.048	SP: 0.464, SC: 0.664 Total: 1.128	SP → SC: Succeed at Round 6 with Rel 68.0% & Pri 0.061	SP: 0.580, SC: 0.651 Total: 1.231
S5	SP → SC: Succeed at Round 6 with Rel 64.0% & Pri 0.040	SP: 0.396, SC: 0.636 Total: 1.032	SP → SC: Succeed at Round 6 with Rel 64.0% & Pri 0.051	SP: 0.495, SC: 0.625 Total: 0.120
S6	SC → SP: Succeed at Round 7 with Rel 76.0% & Pri 0.027	SP: 0.276, SC: 0.756 Total: 1.032	SC → SP: Succeed at Round 7 with Rel 97.0% & Pri 0.028	SP: 0.255, SC: 0.945 Total: 0.945
S7	SP → SC: Succeed at Round 6 with Rel 76.0% & Pri 0.048	SP: 0.456, SC: 0.736 Total: 1.192	SC → SP: Succeed at Round 7 with Rel 91.8% & Pri 0.036	SP: 0.332, SC: 0.890 Total: 1.222
S8	SP → SC: Succeed at Round 6 with Rel 72.0% & Pri 0.056	SP: 0.532, SC: 0.692 Total: 1.224	SP → SC: Succeed at Round 6 with Rel 72.0% & Pri 0.071	SP: 0.665, SC: 0.677 Total: 1.342
S9	SP → SC: Succeed at Round 6 with Rel 68.0% & Pri 0.064	SP: 0.608, SC: 0.648 Total: 1.256	SP → SC: Succeed at Round 6 with Rel 68.0% & Pri 0.081	SP: 0.760, SC: 0.631 Total: 1.391
S10	SP → SC: Succeed at Round 6 with Rel 64.0% & Pri 0.072	SP: 0.684, SC: 0.604 Total: 1.288	SP → SC: Succeed at Round 6 with Rel 64.0% & Pri 0.091	SP: 0.855, SC: 0.585 Total: 1.440

A similar phenomenon also happens in *S3-S10*, where SC serves as a successful offerer in *S6-S7*, while SP serves as one in *S3-S5* and *S8-S9*. However, when both parties switch from concession approaches to tradeoff ones in *S1*, their received utilities are increased together, where SP serves as a successful offerer. In fact, in the ten negotiation scenarios *S1-S10*, one who is good at tradeoff always benefits, and sometime its opponent benefits too. Indeed, for successful offerers, tradeoff approaches outperform concession ones by averagely 23.8% in terms of individual utility.

Observation 2 (*Social Benefit*). Tradeoff approaches produce more social benefit than concession ones.

As an example, let us check *S1*. The *social benefit* (i.e., the total utility) under tradeoff approaches is 0.835 + 0.752 = 1.587, which is higher than that under concession ones (0.668 + 0.748 = 1.416). A similar phenomenon also occurs in *S2-S10*. In fact, in the ten negotiation scenarios *S1-S10*, tradeoff approaches always outperform concession ones by averagely 9.2% in terms of social benefit.

The simulation results for negotiation scenarios in Table III are reported in Table V. It can be seen that both concession and tradeoff approaches succeed in *S11-S19*. As an example, let us check *S12*. Here, SP's and SC's reserved utilities are 0.555 and 0.750, respectively, while their preferred utilities are 0.745 and 0.860, respectively. Under concession approaches, SC accepts the proposal offered by SP at round *six*, with reliability 80.0% and storage price 0.064. Accordingly, their received utilities become 0.596 and 0.756, respectively. It can be verified that their received utilities are greater than their reserved utilities (i.e., 0.596>0.555 and 0.756>0.750), respectively. Under tradeoff approaches, SC accepts the proposal offered by SP at round *eight*, with reliability 82.5% and storage price 0.081. Accordingly, their received utilities become 0.745 and 0.762, respectively. It can be verified that their received utilities are greater than their reserved utilities (i.e., 0.745>0.555 and 0.762>0.750), respectively.

However, tradeoff approaches fail in *S20,* while concession ones still succeed. Here, SP's and SC's reserved utilities are 0.660 and 0.825, respectively, while their preferred utilities are 0.850 and 0.935, respectively. Under concession approaches, SP accepts the proposal offered by SC at round *15*, with reliability 38.0% and storage price 0.068. Accordingly, their received utilities become 0.674 and 0.374, respectively. It can be verified that SP's received utility is greater than its reserved utility (i.e., 0.674>0.660). But, SC's received utility is less than its reserved utility (i.e., 0.374<0.825), which is fine with concession approaches in that SC here serves as the offerer. Under tradeoff approaches, SC fails to make a counter proposal at round *14*. As a result, negotiation breaks down, and each receives utility zero.

Observation 1 also holds here in *S11-S19*. As an example, let us check *S13*. When both parties switch from concession approaches to tradeoff ones, SC's received utility

TABLE V. EXPERIMENTAL RESULTS FOR *S11-S20*

	Concession		**Tradeoff**	
	Offer	**Utilities**	**Offer**	**Utilities**
S11	SP→ SC: Succeed at Round 8 with Rel 86.0% & Pri 0.063	SP: 0.581, SC: 0.811 Total: 1.392	SC → SP: Succeed at Round 9 with Rel 96.3% & Pri 0.072	SP: 0.652, SC: 0.895 Total: 1.547
S12	SP → SC: Succeed at Round 6 with Rel 80.0% & Pri 0.064	SP: 0.596, SC: 0.756 Total: 1.352	SP → SC: Succeed at Round 8 with Rel 82.5% & Pri 0.081	SP: 0.745, SC: 0.762 Total: 1.507
S13	SP → SC: Succeed at Round 6 with Rel 76.0% & Pri 0.056	SP: 0.528, SC: 0.728 Total: 1.256	SC → SP: Succeed at Round 7 with Rel 86.3% & Pri 0.052	SP: 0.482, SC: 0.825 Total: 1.307
S14	SP → SC: Succeed at Round 6 with Rel 72.0% & Pri 0.048	SP: 0.460, SC: 0.700 Total: 1.160	SP → SC: Succeed at Round 6 with Rel 72.0% & Pri 0.061	SP: 0.575, SC: 0.687 Total: 1.262
S15	SP → SC: Succeed at Round 6 with Rel 68.0% & Pri 0.040	SP: 0.392, SC: 0.672 Total: 1.064	SP → SC: Succeed at Round 6 with Rel 68.0% & Pri 0.051	SP: 0.490, SC: 0.661 Total: 1.151
S16	SC → SP: Succeed at Round 7 with Rel 76.0% & Pri 0.036	SP: 0.348, SC: 0.748 Total: 1.096	SC → SP: Succeed at Round 7 with Rel 96.8% & Pri 0.036	SP: 0.327, SC: 0.935 Total: 1.262
S17	SC → SP: Succeed at Round 6 with Rel 80.0% & Pri 0.048	SP: 0.452, SC: 0.772 Total: 1.224	SC → SP: Succeed at Round 7 with Rel 91.6% & Pri 0.044	SP: 0.404, SC: 0.880 Total: 1.284
S18	SP → SC: Succeed at Round 6 with Rel 76.0% & Pri 0.056	SP: 0.528, SC: 0.728 Total: 1.256	SC → SP: Succeed at Round 7 with Rel 86.3% & Pri 0.052	SP: 0.482, SC: 0.825 Total: 1.307
S19	SP → SC: Succeed at Round 6 with Rel 72.0% & Pri 0.064	SP: 0.604, SC: 0.684 Total: 1.288	SP → SC: Succeed at Round 6 with Rel 72.0% & Pri 0.081	SP: 0.755, SC: 0.667 Total: 1.422
S20	SC → SP: Succeed at Round 15 with Rel 38.0% & Pri 0.068	SP: 0.674, SC: 0.374 Total: 0.948	SC → SP: Fail at Round 14	SP: 0, SC: 0 Total: 0

is increased from 0.728 to 0.825, while SP's received utility is decreased from 0.528 to 0.482. Here, SC serves as a successful offerer. A similar phenomenon also happens in *S14-19*. However, when both parties switch from concession approaches to tradeoff ones in *S11*, their received utilities are increased together, where SC serves as a successful offerer. A similar phenomenon also occurs in *S12.* In fact, in the nine negotiation scenarios *S11-S19*, one who is good at tradeoff always benefits. Indeed, for successful offerers, tradeoff approaches outperform concession ones by averagely 19.6% in terms of individual utility. It confirms again that tradeoff approaches produce more utility than concession approaches for one who can offer an attractive proposal to its opponent.

Observation 2 also holds here in *S11-S19.* As an example, let us check *S11*. The social benefit under tradeoff approaches is 0.652 + 0.895 = 1.547, which is higher than that under concession ones (0.581 + 0.811 = 1.392). A similar phenomenon occurs in *S12-S19*. In fact, in the nine negotiation scenarios *S11-S19*, tradeoff approaches always outperform concession ones by averagely 8.7% in terms of social benefit. It confirms again that tradeoff approaches produce more social benefit than concession ones.

However, since tradeoff approaches fail in one scenario (i.e., *S20*) out of ten, their success rate becomes $1-1/10=90\%$. In fact, because of Assumption 1, i.e., incomplete information, it is hard for tradeoff approaches to move towards the preferences of its opponent without miscalculations. By contrast, concession approaches always succeed, and thus their success rate is 100%. The reason is that as both parties employ concession approaches it will reach an agreement eventually. Therefore, tradeoff approaches underperform concession ones in terms of success rate.

VI. CONCLUSIONS AND FUTURE WORK

In Cloud computing, virtualized resources can be reconfigured dynamically to provide elastic services over the Internet. In particular, service consumers are allowed to procure and release resources on demand. As Cloud services are delivered over the Internet, their non-functional properties, which detail how well it is done, matter. However, service consumers and service providers might conflict over non-functional properties. Without negotiation, the conflicts cannot be resolved.

In this paper, we propose tradeoff approaches for Cloud service negotiation. As opposed to concession ones, tradeoff approaches do not reduce one's utility, but still can create a proposal attractive to its opponent. In fact, in order to create such a proposal, one yields on its less important attribute, while demands more on its more important attribute. If its opponent has opposite preferences, the less important attribute surprisingly creates more utility for its opponent. As a result, one eventually moves towards the preferences of its opponent.

Indeed, simulation results show that tradeoff approaches outperform concession ones in terms of both individual utility and social benefit. However, because of incomplete information, tradeoff approaches underperform concession ones in terms of success rate. Therefore, in the future we plan to investigate the game of "chicken" approach for Cloud service negotiation, with an aim to balance utility and success rate. In other words, if one's opponent uses concession strategies, it is better for it to adopt tradeoff ones; if its opponent uses tradeoff strategies, it is better for it to adopt concession ones; if it is uncertain about the strategies of one's opponent, it is better for it to mix the two strategies. In fact, they are the three Nash equilibria for a negotiation game with two pure strategies. Moreover, we plan to extend Algorithms 1 and 2 to handle more than two attributes, i.e., multiple attributes, and to relax Assumption 2 to deal with nonlinear relationships between the values of attributes and their corresponding utilities. Third, we plan to incorporate the time factor, which plays a critical role in real-world negotiations, into the negotiation process.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable comments on the paper. The research is supported by the Social Sciences and Humanities Research Council of Canada (SSHRC) and the Natural Sciences and Engineering Research Council of Canada (NSERC).

REFERENCES

[1] L. M. Vaquero et al., "A Break in the Clouds: Towards a Cloud Definition", *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 1, 2009.

[2] M. Armbrust et al., "A View of Cloud Computing", *The Communications of ACM*, vol. 40, no. 4, pp.50-58, 2010.

[3] K. M. Sim, "Towards Complex Negotiation for Cloud Economy", In *R. S. Chang et al. (Eds.): GPC 2010, LNCS 6104*, pp.395-406, 2010.

[4] I. Brandic, D. Music, and S. Dustdar, "Service Mediation and Negotiation Bootstrapping as First Achievements towards Self-Adaptable Grid and Cloud Services", In *Proc. of International Workshop on Grids Meets Autonomic Computing (GMAC 2009)*, Barcelona, Spain, 2009.

[5] P. Faratin, C. Sierra, N. R. Jennings, "Negotiation Decision Functions for Autonomous Agents", *International Journal of Robotics and Autonomous Systems*, vol. 24, no. 3-4, pp.159-182, 1997.

[6] N. R. Jennings et al., "Automated Negotiation: Prospects, Methods and Challenges". *Group Decisions and Negotiation*, vol. 10, no.2, 2001, pp.199-215, 2001.

[7] A. Lomuscio, M. Wooldridge, "A Classification Scheme for Negotiation in Electronic Commerce", *Group Decision and Negotiation*, vol. 12, no. 1, pp.31-56, 2003.

[8] D. Besanko and R. R. Braeutigam, "Microeconomics", 3rd ed., John Wiley & Sons Publisher, Hoboken, NJ, USA, 2008.

[9] S. Ferretti et al., "QoS-aware Clouds", In *Proc. of the 3rd International Conference on Cloud Computing (CLOUD 2010),* Miami, USA, 2010.

[10] A. Chazalet, "Service Level Checking in the Cloud Computing Context", *In Proc. of the 3rd International Conference on Cloud Computing (CLOUD 2010)*, Miami, USA, 2010.

[11] H. Li and J. J. Jeng, "CCMarketplace: A Marketplace Model for a Hybrid Cloud", In *Proc. of the 20th Conference of the Center for Advanced Studies on Collaborative Research (CASCON 2010),* Toronto, Canada, 2010.

[12] A. Rubinstein. "Perfect equilibrium in a bargaining model", *Econometrica*, vol. 50, pp.97-110, 1982.

[13] S. D. Conte et al. "Software Engineering Metrics and Models", The Benjamin/Cummings Publishing Company, Inc., Menlo Park, CA, USA, 1986.