Using Resource-level Information into Nonadditive Negotiation Models for Cloud Market Environments

Mario Macías*†
*Barcelona Supercomputing Center
Jordi Girona 29
08034 Barcelona, Spain
{mario.macias,jordi.guitart}@bsc.es

Jordi Guitart*†
†Technical University of Catalonia
Jordi Girona 1-3
08034 Barcelona, Spain
{mario,jguitart}@ac.upc.edu

***Abstract*—Markets arise as an efficient way of organising resources in Cloud Computing scenarios. In Cloud Computing Markets, Brokers that represent both Clients and Service Providers meet in a Market and negotiate for the sales of resources or services. This paper defends the idea that efficient negotiations require of the usage of resource-level information for increasing the accuracy of negotiated Service Level Agreements and facilitating the achievement of both performance and business goals. A negotiation model based on the maximisation of nonadditive utility functions that considers multiple objectives is defined, and its validity is demonstrated in the experiments.**

I. Introduction

In the recent years, the big mainframes paradigm where users own their computing resources is being progressively transiting to a more utility-driven paradigm, where users do not own their resources and pay for the usage of remote resources [1]. Cloud Computing [2] is the most promising current implementation of Utility Computing in the businesses. With this new evolution, the classic Resource Management mechanisms became inefficient because some reasons such as the heterogeneity and dispersion of resources.

Market-based resource management [3] is proposed as a paradigm to deal with the complexity because the possibility of doing business will motivate Service Providers to offer their resources in the system and give a Quality of Service (QoS) according to their real capacity. In addition, Market mechanisms obligate the users to adjust their reservations of the system to their real space and time requirements. Another advantage is that it is relatively easy to implement in a decentralised architecture, where participants enter in the Market looking for the satisfaction of their own necessities, and they do not need to know about the global status of the system to maximise their utility.

In Market-based Cloud computing, either client applications or end users that want to use remote resources or services, and providers that want to sell their services, contact with economic agents (called Brokers) which will negotiate with other Brokers to buy/sell the services in a Market. When the Client Brokers find their requirements in the Market, a **negotiation process** is started to establish the terms of the contract. If both parts reach an agreement, the terms of the contract are specified in a Service Level Agreement (SLA) and the Client's application can use the resource. During the usage of the resources, the correct fulfilment of the terms of the SLA is watched by a neutral entity, and penalises the buyers or the sellers if they violate the SLA.

Since Brokers that negotiate for the sales of services are autonomous, it is needed to provide them with some business models and intelligent behaviour so they are able to take the best decisions for Client applications or Service Providers, and maximise their utility. Current proposals on utility function models for Market-based Utility Computing negotiations [4], [5], [6] are additive: they assume that all the factors of the negotiation are independent from the others. This paper defends the idea that in real negotiations, not all the terms under negotiation are independent from each others, and proposes the usage of non-additive utility functions, where the terms can be interdependent (see section III-B).

This paper enhances existing business models for negotiation and applies them to the sales of services between computing agents: when a Provider Broker negotiates an SLA with a Client Broker, it takes into account some economic terms such as price, but also technical parameters such as Quality Of Service (QoS) that have influence in the economic terms: for a purely-economical Provider Broker, it is very difficult to quantify the SLOs, since it has not enough technical knowledge about the status and punctual capacities of the resources. The components described in this paper use resource information in negotiation time to, for example determine if a task can be executed or not, or the minimum price to make this task profitable for the Provider. According to this, the main contributions of this work are:

1) Modelling and characterisation of the negotiations required to perform sophisticated sales in Market-Based Cloud Computing in function of the desirable objectives, by using nonadditive utility functions.
2) Evaluation of the proposed business models for the negotiation between Brokers. This includes the comparison of several values for the parameters of the model and the evaluation about its feasibility and influence in the achievement of desired objectives.
3) Usage of low-level dynamic knowledge, provided by the resource fabrics, for supporting economic negotiations. The required knowledge is defined by the contributions enumerated in point 1 and 2, and is acquired by the

resource fabrics monitor in real time.

The remainder of the paper is organised as follows: Section III describes the models used for the negotiation; Section IV describes the evaluation environment and the results of the simulation; finally, Section V shows the conclusions of this work and points to some future research lines.

II. Related Work

This paper is based in the work performed within the SORMA European project [7], concretely in the Economically Enhanced Resource Manager (EERM) [8], [9] component. EERM combines the purely economic knowledge (because is in direct contact with the economic layers of SORMA Marketplace) and the plain resources data (because it manages directly the resource fabrics) to help Brokers to perform better negotiations and enforce the resource management, not only having into account performance but also economic goals.

Raiffa [6] established and compiled the mathematical basis of the negotiation models. He classified the different negotiation models in base to the characteristics of the environment and the negotiated goods. It will be widely referenced in this paper, as in the most of the other works about negotiation.

Faratin et al. [5] applied and extended some existing models for service-oriented decision functions in bilateral negotiations between autonomous agents. It concentrates in many-parties, many-issues, single-encounter negotiations with an environment of limited resources. Since computing services are qualitative in nature rather than quantitative, Faratin extends this model by adding qualitative values and associates fuzzy sets to them in order to express better the quality in the negotiations.

Once the agents have determined the set of variables over which they will negotiate, the negotiation process between two agents consists of an alternate succession of offers and counteroffers of values for the x, until an offer or counteroffer is accepted or rejected by the other side or one of the parties terminates the negotiation. Faratin et al. demonstrated what this paper affirms: negotiation tactics must be responsive to changes in the environment.

The work in this paper tries to extend the model of Faratin by extending the information extracted from the resources and used in the negotiation, and by having into account other economic factors, such as reputation, risk management, etc. Another difference is that Faratin's work is limited to a particular scenario: Client and Provider Brokers meet in an isolated way to negotiate for a single type of resource. This paper extends this scenario to consider a Market place where the Brokers negotiate for a huge range of services.

Venugopal et al. [10] introduces a bilateral negotiation protocol similar to the exposed in this paper, based in the Alternate Offers mechanism. The main difference is that the offers/counteroffers cycle of this paper is predefined finite, and in the Alternate Offers it can continue indefinitely until one of the parts decides to stop the negotiation. This paper discarded this possibility mainly by efficiency questions.

III. Characterisation of the Negotiation

A. Negotiation protocol

Before the negotiation starts, the EERM of the Service Provider must register its offered services into the Market, by providing some semantic information that allows identifying what service is and its functionalities, and an extra meta-SLA with some data about the SLOs that the Service Provider is willing to negotiate.

When a Client Broker wants to acquire a service, it queries the Market by providing some semantic information, and gets a list of the Service Providers that match the requirements (every Provider has its own EERM) and the meta-data about the negotiable SLA terms. After that, it selects the suitable Providers, and creates a proposal of agreement for each one; using the meta-data it creates an uncompleted SLA with its requirements, and leaves other SLOs blank. When the EERM receives the SLA proposal, it evaluates if the proposed terms can be accepted. If the Client Broker received from the EERM an acceptance message or a counteroffer, it evaluates it and finishes the negotiation by rejecting the SLA or by sending a confirmation message to the EERM.

This work was performed within the SORMA project, which provides interfaces and protocols for the registration in the market and the negotiation of resources. However, our proposal does not rely on any particular interface specification, since the work is focused exclusively on the negotiation models and resource management. Although the model has been designed to work within a Market framework, it is applicable to the most of current commercial Cloud providers, even if they do not implement market support. It is only necessary to have some providers that want to sell their resources freely on the Internet. In this scenario, the client brokers only have to know the endpoint where the brokers of the providers are, to start a negotiation without any intervention of a market. The nonexistence of a market would decrease the economic efficiency of the system, because the competence, the discovery and the reputation mechanisms would be limited.

B. On the usage of non-additive utility functions

First, we must define the analytic model for representing the negotiations that will be performed by the EERM. This model must take into account the negotiated SLOs and other terms, such as Client classification or reservation slots plus the sale price.

Usual negotiation models for Utility computing are based in the models proposed by Raiffa [6] and Faratin [5]. This model is pretty easy to manage and calculate the maximum and minimum utilities. However, it is an *additive model* which assumes that all the factors are independent from the others.

Let S be the SLA under negotiation, Equation 1 shows the general form of the *nonadditive utility function* U used in this paper from the Service Provider side.

$$U(S) = \sum_{i=1}^{m} o_i u_i(S) \tag{1}$$

Where m is the number of goals for the Provider, such as revenue maximisation, reputation, performance maximisation, high occupation of resources, or satisfaction of certain type of users. u_i is the sub-utility function that defines how much will be the objective i satisfied, and o_i is a number between 0 and 1 that defines the priority that the Provider assigns to the particular objective. It must be considered that $\sum_{i=1}^{m} o_i = 1$.

Although Equation 1 is similar to an additive function, actually it is not. Instead of calculating each of the sub-utility functions as a function of a single SLA term and finally add them up, Equation 1 calculates all the sub-utilities as a function of the whole SLA, because the different objectives are not independent from the others and, for example, revenue maximisation can affect negatively the Client satisfaction.

C. Negotiation terms and Utility functions

In this section, we define the set O of objectives, the set S of SLA terms and the utility function $U(S)$ that calculates how beneficial the proposed SLA is for the objectives of the Provider. Prior to this, some representative objectives for typical Service Providers must be chosen. In this paper, four objectives have been chosen: $O \subseteq \{o_{rv}, o_{cc}, o_{ph}, o_{rp}\}$.

- o_{rv} is the objective that defines the maximisation of the revenue. The most common motivation of most of the Providers that enter in the Market is to earn money by selling its resources. So the higher is the global revenue, the higher is u_{rv}.
- o_{cc} is an objective used for Client classification [11]. This gives preference to the local users (or users from a near organisation) over the non-related users.
- o_{ph} is the objective that gives preference to tasks or services to be executed in off-peak hours, to prevent the system overload during peak hours.
- o_{rp} is the objective used for maximising the reputation of the Provider [12]. Reputation is an important issue, because if a Provider violates a SLA it not only must pay a penalty to the Client, but also will loss its reputation in the Market. In consequence, future Clients will be reticent to buy their resources.

Literature details revenue as the main objective in markets. The other objectives where chosen as examples of suitable goals in such a scenario according to the related work. However, each provider must decide his own relevant goals.

This paper demonstrates how the behaviour of the Provider can be modulated by changing the values of the components of O that multiply their associated sub-utility functions $u_{rv}, u_{cc}, u_{ph}, u_{rp}$ in negotiation time, as can be shown in the utility function applied to the context of the chosen objectives (Equation 2).

$$U(S) = o_{rv}u_{rv} + o_{cc}u_{cc} + o_{ph}u_{ph} + o_{rp}u_{rp} \tag{2}$$

The rest of this section describes and justifies the sub-utility functions chosen in this paper, calculated in base to the SLA $S \subseteq \{M, C, CP, Rev, \Delta t\}$, where M, C are the Memory and CPUs amount to acquire, $0 \leq CP \leq 1$ is the indicator of Client Priority, Δt is the time slot where the resources are assigned and Rev is the revenue acquired by the sale. All the sub-utilities are normalised to the same range $[-1, 1]$ because otherwise the influence of the weights O would be distorted by the differences between the ranges of the sub-utilities.

1) Price maximisation: Before describing u_{rv}, u_{cc}, u_{ph} and u_{rp}, it is advisable to describe an utility function that it is not used as a term of $U(S)$, but some of the other sub-utility functions depend on it: the price maximisation utility function.

When the Provider proposes a price, it must know the range of prices where the agreement is possible. The *reservation price of the seller* (RP_s) is the minimum price that the seller can accept without losing money. The *reservation price of the buyer* (RP_b) is the maximum price it can pay and still being beneficial for its objectives. An agreement between buyer and seller is only possible when $RP_s \leq Price \leq RP_b$.

Equation 3 defines the utility for given revenue:

$$u_p(S) = \frac{Price - RP_s}{RP_b - RP_s} \tag{3}$$

That means that the utility of the price for the Provider is higher (~ 1) when the revenue of the Provider tends to be RP_b. However, this sub-utility function is not used directly as a term in $U(S)$, because in a competitive Market high prices will enforce Clients to look for cheaper Providers.

The main issue of implementing this formula is to know the reservation price of the buyer, which only can be speculated in function of the Market history.

2) Total revenue maximisation: For maximising the total revenue, it is needed to have into account the price of the sale, but also the status of the competitive Market. Having into account the *Law of Supply and Demand* [4], it is needed to define u_{rv} to propose different prices in function of the Market status, so they will tend to be higher in demand excess scenarios and lower in offer excess scenario.

To check the Market status, an **aggressiveness factor** $a(t)$ has been defined: it is intended to limit the profit expectations of the provider. In scenarios where many clients ask for scarce resources, providers are in an opportunistic situation for asking prices that are near to the buyers' reservation price. In this case, $a(t) \to 1$. In the opposite scenario (offer excess) the providers must limit their economic pretensions, so $a(t) \to 0$.

Let t be the current time, H be the length of an historic time period, $C_{tot}(t)$ be a constant function whose value is the number of CPUs of the Provider, and $C_{used}(t)$ be a function that describes the number of busy CPUs in the Provider over time. Equation 4 is the *ideal aggressiveness factor* $a'(t)$.

$$a'(t) = \frac{\int_{t-H}^{t} C_{used}(t)\,dt}{\int_{t-H}^{t} C_{tot}(t)\,dt} \tag{4}$$

Let $0 \leq \delta \leq 1$ the *aggressiveness adjustment rate* which shows how quick the actual aggressiveness will tend to the ideal aggressiveness, Equation 5 shows how the aggressiveness $a(t)$ is adjusted in function of the ideal aggressiveness and the previous actual aggressiveness:

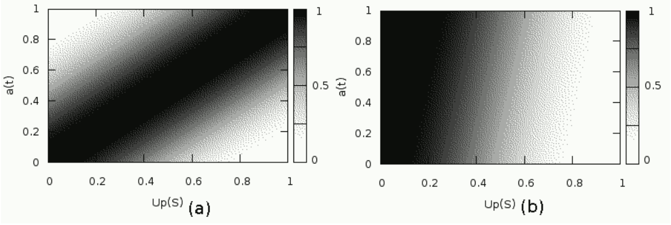

Figure 1. Firt attempts for defining $u_{rv}(S)$

$$a(t) = a'(t)\delta + a(t-1)(1-\delta) \tag{5}$$

Once $a(t)$ are defined $u_p(S)$, there is enough data to define an $u_{rv}(S)$ to achieve the next goals:

- In an offer excess scenario, where $a(t)$ is low, Clients will choose Providers that offer lower prices ($u_p(S) \to 0$) for the same SLA. So $u_{rv}(S) \to 1$ when $u_p(S) \to 0$.
- In a demand excess scenario, where $a(t)$ is high, Clients will have to accept high prices ($u_p(S) \to 1$), since they have very few alternatives. So it is convenient for the Provider to push up its prices for maximising its benefit.

First intuition says that the Law of Supply and Demand can be accomplished by adjusting linearly the prices in function of the demand, as shown in the maximum values (darkest colour) of Figure 1(a). The equation that describes this behaviour is $u_{rv}(S) = \sin\left(\frac{\pi}{2}\left(u_p(S) + (1 - a(t))\right)\right)$. However, the experimentation results shown that, even if $a(t)$ is relatively high, the Clients have chances to choose cheaper Providers, so maximising $u_{rv}(S)$ would lead to have less revenue.

Alternatively, one can divide $a(t)$ in order to decrease the utility when prices are too high in high-demand market scenarios. The new formula is $u_{rv}(S) = \sin\left(\frac{\pi}{2}\left(u_p(S) + (1 - a(t)/div)\right)\right)$, where div denotes the slope of the crest of the function maximums, as can be seen in Figure 1(b). That function is effective in normal Market status, but not in those where the demand excess is extremely high ($a(t) \simeq 1$), because it does not take profit from the good position of the Provider in the negotiation.

First attempt (Figure 1(a)) is precise when $a(t)$ is low, but not when $a(t)$ is high. However, the results of second attempt (Figure 1(b)) are opposite to the first one. To try to combine both, instead of dividing $a(t)$, in the third attempt $a(t)$ has been powered to $curv$ (see Equation 6), that describes the intensity of the curve of the crest of Figure 2. In this way, the prices will be low in almost all the scenarios, excepting in the very excess of demand, when $a(t) \to 1$ and the prices can be high. In addition, $u_p(S)$ is multiplied by an attractor called G that will make utilities lower when combinations of $(u_p(S), a(t))$ are far from the crest of the function. That will force even more providers to look for combinations $(u_p(S), a(t))$ near to the maximum of the utility.

$$u_{rv}(S) = \sin\left(\frac{\pi}{2}\left(G \cdot u_p(S) + (1 - a(t)^{cur})\right)\right) \tag{6}$$

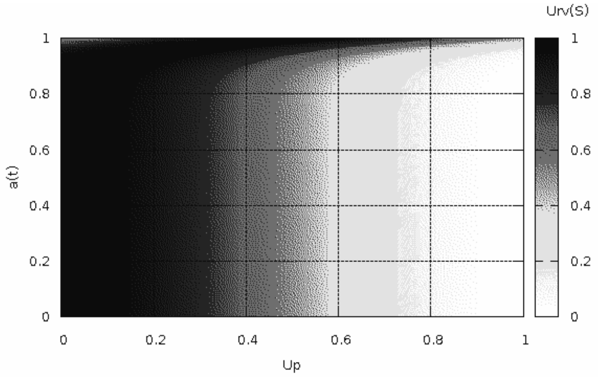

Figure 2. Colour map that represents the value of $u_{rv}(S)$ in function of $u_p(S)$ and $a(t)$

Since the range of utilities in Equation 6 is $[-1, 1]$, the whole equation is divided by 2 and added 0.5 to normalize the range of utilities to $[0:1]$. The resulting formula is the Equation 7.

$$u_{rv}(S) = 0.5 + \frac{\sin\left(\frac{\pi}{2}\left(G \cdot u_p(S) + (1 - a(t)^{cur})\right)\right)}{2} \tag{7}$$

After several tests and experiments in competing market simulations, the combinations of values for G and cur constants that provide the best results are $G = 2$ and $cur = 15$, which are used in the experiments performed in the paper.

The colour map in Figure 2 helps to understand better the function in Equation 7. The dark zones show these combinations of $u_p(S)$ and $a(t)$ that gives higher values for u_{rv}.

3) Client classification: Client classification is performed through price discrimination [11]. The parameter CP is the Client Priority, which tends to 1 when the Client is much related to the organisation of the Provider, and tends to 0 when there is absolutely no relation between the Client and the Provider. It is calculated as the Euclidean distance between Client and Provider in a multi-dimensional space.

Equation 8 is used to define the utility for Client classification. Given u_{rv} and CP, if the Client priority is high, the utility will be higher when u_{rv} is low (the Provider must not be expensive for related Clients). If the Client priority is low, the utility will be higher when u_{rv} is high.

$$u_{cc}(S) = \begin{cases} CP + u_p(S) & \text{if } u_p < 1 - CP \\ 2 - CP - u_p(S) & \text{otherwise} \end{cases} \tag{8}$$

4) Prioritisation of off-peak hours: Let $\Delta t = t_f - t_i$ be the interval of time where the task is executed, $C_{tot}(t)$ be a constant function whose value is the number of CPUs of the Provider, $C(t)$ be a constant function whose value the number of CPUs requested to the task under negotiation, and $C_{used}(t)$ be a function that describes the number of busy CPUs in the Provider over time. Equation 9 is the utility function that is higher when more resources are free, and near 0 when the Provider resources are near its maximum occupation.

$$u_{ph}(S) = 1 - \frac{\int_{t_i}^{t_f} C_{used}(t) + C(t)\, dt}{\int_{t_i}^{t_f} C_{tot}(t)\, dt} \tag{9}$$

In the experiments, both CPU and Memory are negotiated. But since CPU is the bottleneck, it is used as the resource for calculating the peak hours.

5) Utility for reputation: Let R_0 be the reputation of the Provider in negotiation time, R_v be the future reputation of the Provider in case of SLA violation, R_f the future reputation in case of correct SLA fulfilment, and P the probability of violating an SLA (calculated from historical data); Equation 10 shows the utility of keeping the reputation of the Provider:

$$u_{rp}(S) = \frac{P\, R_v + (1-P)R_f}{R_0} - 1 \tag{10}$$

R_v and R_f are calculated as described in [12].

D. Maximising the utility function

When the Provider receives an offer, it must specify a price and a range of time (if the requested time is not fixed) to maximise the utility function.

Maximising nonlinear utility functions can be pretty complex, especially when multiple variables exist. Choquet Integrals [13] have been used for multi-criteria decision with nonadditive functions where some of their values are fuzzy. However, they do not help to maximise the function, but only to choose the best alternative from a set.

The framework used in this paper uses discrete values of time and price, and does not need fuzzy logic because the data used in the utility functions is well known by the Provider (excepting the Reservation Price of the Client, which is speculated). Then $U(S)$, which is theoretically continuum, is divided into a finite set of values in function of discretised price and time. Choosing the best price and time slot is choosing the pair of price and time whose $U(S)$ is greater to the $U(S)$ values for all the other pairs.

E. Why is the resource information needed?

When calculating $u_p(S)$ for price maximisation, it is important to know the status of the resources and how an incoming SLA can affect into this status, in order to quantify them economically and calculate the Reservation Price of the Seller.

$u_p(S)$ and the $a(t)$ have a decisive role when calculating the utility for maximising the global revenue $u_{rv}(S)$. Section III-C shown how $a(t)$ is calculated as a function of the historical monitoring data from the resources. The same historical data is also used to calculate $u_{ph}(S)$, that gives more importance to the jobs which are located in off-peak hours.

The resource information is also really important when calculating $u_{rp}(S)$, because the probability P of breaking an incoming SLA is calculated in function to statistical monitoring data of past executions and the current monitoring status.

Even an utility function such as $u_c c(S)$ used to perform Client Classification, has relation with the resource information, since it is calculated as a function of $u_p(S)$.

IV. Evaluation

A. Simulation environment

A simple Market has been simulated to test the validity of the negotiation model. Client Brokers that represent either a Web or Grid Client enters in the Market to ask for web workload or for plain resources. The workload for Grid has a random distribution and the workload for Web Services has distribution taken from a real Web application, with variable workload in function of the hour of the day [14].

Grid Clients send an SLA proposal where is specified the plain resources (CPU and Memory) to buy, the duration of the job, and a time interval where the job can be executed (bigger than the duration, to let the EERM schedule the best execution time). Web Service Clients send a required workload for a service, and a fixed time interval to use the services (there is no arbitrary schedule of the reservation, since Web users want the services for the same moment).

Both Client types also must specify what QoS class they want: Gold, Silver, or Bronze. Gold Clients will pay the triple than Bronze Clients, and Silver Clients the double than Bronze ones. The average failure rates for Gold, Silver, and Bronze services are, respectively, 0.5%, 1%, and 2%.

A Client looks for potential Providers in the Market and sends SLA Proposals to all of them. After that, the Providers accept/deny the proposals and return to it a time allocation and a price, based on the maximisation of their utility function. Finally, the Client chooses the Provider with a best price and time schedule for its interests and sends it a confirmation.

The Provider can violate the SLA due to an internal error, or because it receives a proposal from another Client that can not be allocated but is interesting to accept it and cancel the other (it is decided by the utility function having into account objectives such as Client classification or Revenue maximisation) [9], [8]. This violation will affect to the reputation of the Provider, which is taken into account by the Client in negotiation time: when choosing the best SLA, the price proposed by the Provider is divided by its reputation, so the Client will consider the price of a Provider with low reputation higher than the same price from a Provider with high reputation.

B. Checking statistically the effectiveness of the sub-utility functions

In this test, repeated simulations are performed in a competitive Market with 100 Providers, whose different objective weights for $U(S)$ are generated randomly to provide some statistically valuable data. Simulations are repeated with a number of Clients that vary from 50 (offer excess) to 1000 (demand excess). Each Provider is selling 20 CPUs and 6GB of RAM memory.

This section shows the results of the simulation in terms of the four objectives described in Section III-C. For each simulation, the next data is collected from the Provider side:

- *Revenue*: total revenue of the Provider.
- *AvgPrice*: average price of Resource/Hour sold.

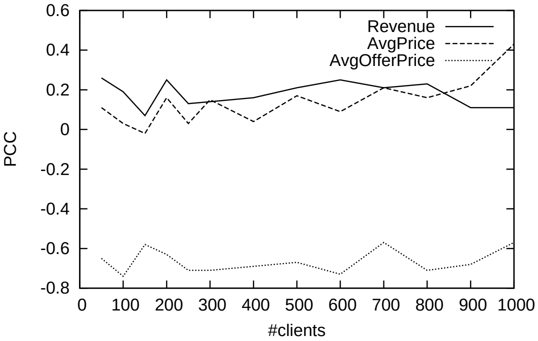

Figure 3. Correlation between o_{rv} and some output parameters

- *AvgAffinity*: average affinity of Clients that use the system (value CP of section III-C3).
- *AvgReputation*: average reputation of the Provider during the whole execution.
- *AvgOfferPrice*: the price of Resource/Hour that the Provider offers to the Client. The difference with *AvgPrice* is that it includes only the prices of the agreed negotiations, and *AvgOfferPrice* includes the offer prices for both the agreed and non-agreed negotiations.

To show the effectiveness of the utility functions proposed in this paper, the Pearson Correlation Coefficient (PCC) [15] between the collected data and the objectives o_{rv}, o_{cc}, o_{ph} and o_{rp} is calculated.

1) Revenue maximisation: Figure 3 shows how o_{rv} has a slightly influence in the total revenue of the Provider (around 0.2). Obviously the correlation coefficient cannot be 1 because there are many other factors that have influence in the revenue. However, there is a big negative linear relationship between o_{rv} and the price that the Provider proposes for the sale of the resource in negotiation time (AvgOfferPrice): Providers that want to sell more must decrease their prices. However it can be observed that o_{rv} has a positive influence on the prices of the sold resources (AvgPrice). It is because the maximisation of $U(S)$ will lead to ask the optimum prices in function of the Market status (speculated by $a(t)$).

In a extreme demand excess scenario (900-1000 Clients), where the Provider can be more aggressive in its negotiations, it can be observed a positive correlation between o_{rv}, AvgOfferPrice and AvgPrice. However the correlation with the total revenue is more or less the same. That does not mean that the utility function is less efficient, it means that all the Providers increase their revenues because the Market status, and the influence of o_{rv} in the total revenue is less in percentage.

2) Client classification: Figure 4 shows the effectiveness of the inclusion of $o_{cc}u_{cc}(S)$ in the general utility function: the higher is o_{cc}, the higher is the affinity (around 0.6 in all the Market scenarios).

As described in Section III-C3, u_{cc} is strongly related with u_p. Figure 4 reflects this relation as a negative correlation between the o_{cc}, the global revenue, and the average price. Since the Provider will try to sell to affine customers, it will offer them its resources at lower prices, and there are more possibilities that Clients choose affine Providers.

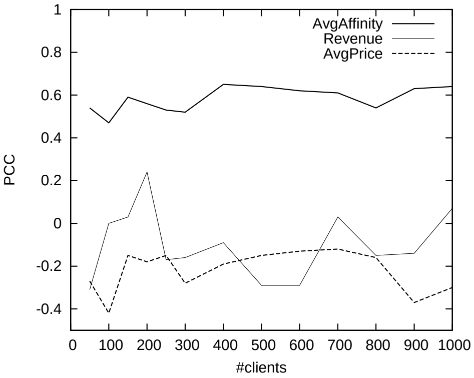

Figure 4. Correlation between o_{cc} and some output parameters

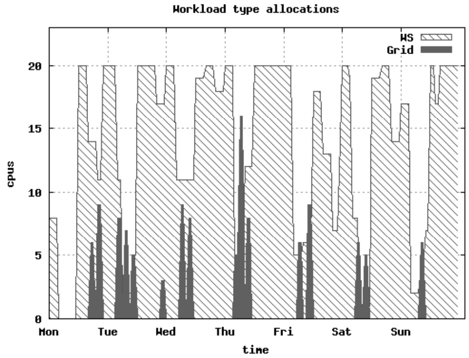

Figure 5. Allocation in time of workloads divided by Web Services or Grid

3) Priorisation of off-peak hours: Web Services must be executed in fixed intervals, and Grid jobs have a random distribution, but since they are not real-time applications, they can be scheduled to be executed in the future. Figure 5 shows how the inclusion of $o_{ph}u_{ph}(S)$ in $U(S)$ allows the Providers offering better prices to the Clients in off-peak hours and, in consequence, the Grid jobs are automatically executed when the Web Services workload is low.

4) Reputation: Figure 6 shows that, unlike we expected before running the simulations, o_{rp} does not have any influence on the average reputation of the Provider. However, the results are interesting because u_{rp} acts as a risk manager. The figure show how the average price of the sold resources is increased or decreased in function to the reputation. This means that the Provider will charge a small amount of money to compensate possible losses as consequence of the loss of reputation.

Previous work [12] shows clearly the importance of keeping a high reputation. In the experiments, the revenue increases almost linearly with the reputation. At equal prices, a Client will choose the Provider with higher reputation. The alternative to Providers with low reputation is to decrease their prices.

C. Comparison with fixed-pricing Providers

In this experiment, 10 different Providers have been compared: four Providers that implement negotiation as proposed

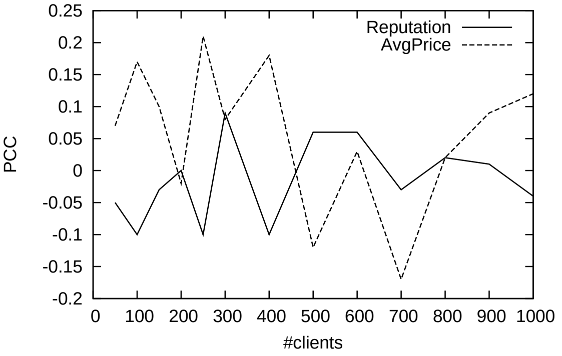

Figure 6. Correlation between o_{rp} and some output parameters

in this paper and six Providers that implement fixed pricing. Each one of the Providers that use nonadditive utility functions for negotiation has a main objective whose o weight value is 0.55 and the other secondary objectives have a weight of 0.15. So there is a Provider that prioritises revenue maximisation, other that prioritises Client classification, one that prioritises off-peak hours, and other that prioritises reputation maximisation. On the side of Providers with fixed pricing, since it is difficult to know beforehand what is the best fixed price, six Providers with different prices have been added into the testbed. Each Provider proposes always a fixed percentage between the RP_S and the RP_b, by having respectively fixed values of $\alpha = \{0.04, 0.06, 0.08, 0.10, 0.12, 0.14\}$ in the next pricing formula:

$$Price = RP_s + \alpha \left(RP_b - RP_s \right)$$

Since in a competing Market fixed-pricing Providers do not have influence in the peak minimisation nor the reputation maximisation decisions of their competitors, only Providers that maximise revenue and perform the Client classification have been compared with the fixed-pricing ones. Furthermore, the influence of not implementing peaks minimisation in them is shown. However, the influence of policies for reputation maximisation cannot be shown because the data set in this experiment is too small to establish correlations and the influence of $o_{rp}u_{rp}(S)$ cannot be compared in terms of achieved reputation with the other Providers.

1) Revenue maximisation: Figure 7 shows that adaptive pricing by maximisation of nonadditive utility functions is the best choice in almost all the scenarios. In the high excess of offer scenarios (20 Clients), fixed-pricing Providers that sell their resources at low price (Fix 4% and Fix 6%) have more revenue than adaptive pricing Providers (RV). This is because the biggest part of the demand is shared across Providers with low prices, and since these Providers sell their resources at higher prices than RV, they earn more money.

Also in the highest demand scenarios (8 to 10 Clients per each Provider), the fixed-pricing Providers with highest prices earn more money than the adaptive pricing one. In this case, Clients do not have enough alternatives for choosing and they must accept almost any offer. Providers with highest prices can take advantage of this situation.

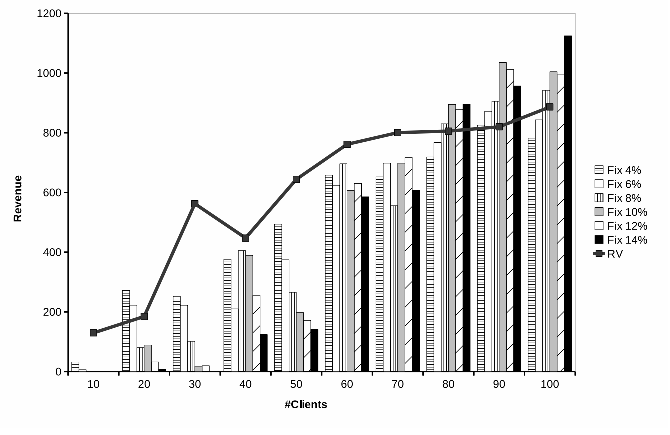

Figure 7. Comparison of revenue between a Provider that tries to maximise the revenue (RV) with fixed-pricing Providers

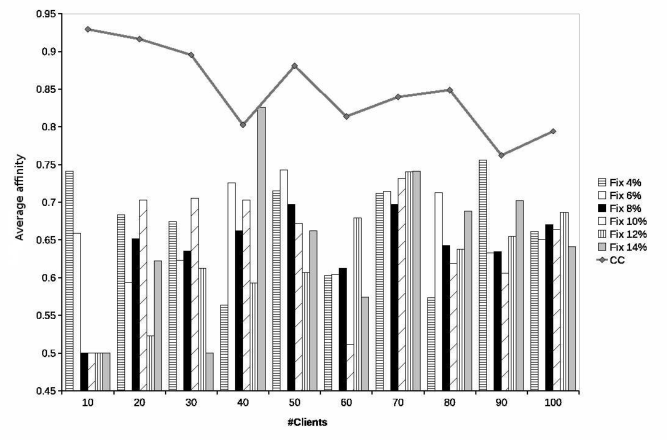

Figure 8. Comparison of Client affinity between a Provider that performs Client classification (CC) with fixed-pricing Providers

2) Client classification: Figure 8 shows in a graphical way how efficient is Client classification compared to Providers that do not perform it (the fixed-pricing ones).

It can be seen how average affinity of Clients decreases when the Clients number increases. It is because the Provider with Client classification can accept almost all the affine users when its resources are idle, but when it is overloaded, most of the Clients cannot use its resources and must look for less affine ones. A way of keeping high affinity in all the scenarios could be by implementing job cancellation for low-affinity users. But this solution will entail other important problems, such as economic losses due to the pay of penalties and the loss of reputation.

3) Peak-hours minimisation: Figure 9 shows clearly the influence of not having policies for the minimisation of off-peak hours. Since the Providers that use nonadditive utility functions maximisation can allocate the workloads in off-peak hours at better prices, (as shown in figure 5), Providers that do not implement peak minimisation policies do not execute Grid

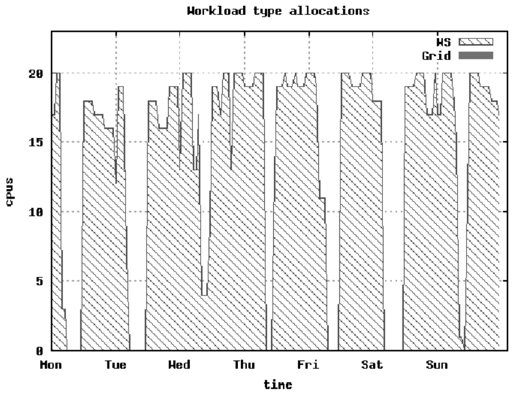

Figure 9. Sample distribution of workload types in a Provider without peak-hours minimisation policies. It can be observed that there are no Grid workloads

workloads. In the experiments performed, only the Provider that offered the lowest fixed prices (4% over RP_s) in demand excess scenario executed 3 Grid jobs. The allocations for all the other fixed-pricing Providers do not include any Grid job.

V. Conclusions and Future Work

The first aim of this paper is to show the benefits of applying knowledge about resources to Market negotiations. In Cloud Computing Markets, this will lead to the Market Broker to perform better business decisions. This paper demonstrated how the figure of an EERM can benefit Providers by providing resource data to the Brokers and by considering economic policies in resource management.

The other contribution of this paper is the intention of being a step forward in the modelling and evaluation of utility functions for negotiations in Cloud Computing Markets. The simulations show how a Provider can perform complex actions by only maximising a multi-dimensional utility function. The contribution of these experiments is based in the usage of nonadditive utility functions, more difficult to treat, but needed when assuming that the terms under negotiation are not independent between them. In the model defined in Section III-C, the utilities for Client classification $u_{cc}(S)$ and revenue maximisation $u_{rv}(S)$ were related by the price: maximising the price would lead to maximise $u_{rv}(S)$, but to minimise $u_{cc}(S)$ for affine Clients.

The proposed nonadditive utility function considers the possibility of having multiple objectives in a same entity, such as revenue maximisation, Client classification, reputation or load-balancing in time. In order to keep the efficiency both in business and performance terms, most of the parameters that compose the utility function are collected dynamically from the resource-level information. This paper has shown the high importance of having this information available in negotiation time. The simulations performed demonstrate how the objectives can be partially achieved by balancing correctly their weights in the utility function.

This paper leaves some open lines for future research, such as improving utility functions for more efficient negotiations and extending their terms to include other economic or performance goals, finding methods for the maximisation of complex nonadditive utility functions that include fuzzy values for nondeterministic data, and evaluating the validity of the model in a real Cloud Computing Market, taking real data from the resource fabrics and compare it with other existing models.

Acknowledgment

This work is supported by the Ministry of Science and Technology of Spain and the European Union (FEDER funds) under contract TIN2007-60625, by the Generalitat de Catalunya under contract 2009-SGR-980, and the Commission of the European Communities under IST contract 034286 (SORMA).

References

[1] M. A. Rappa, “The utility business model and the future of computing services,” *IBM Syst. J.*, vol. 43, no. 1, pp. 32–42, 2004.

[2] R. Buyya, C. Yeo, S. Venugopal, J. Broberg, and I. Brandic, “Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility,” *Future Generation Computer Systems*, vol. 25, pp. 599–616, 2008.

[3] C. S. Yeo and R. Buyya, “A taxonomy of market-based resource management systems for utility-driven cluster computing,” *Softw. Pract. Exper.*, vol. 36, no. 13, pp. 1381–1419, 2006.

[4] J. Guitart, M. Macías, O. Rana, P. Wieder, R. Yahyapour, and W. Ziegler, *Market-Oriented Grid and Utility Computing*. Wiley, 2009, no. 12, ch. SLA-based Resource Management and Allocation.

[5] P. Faratin, C. Sierra, and N. R. Jennings, “Negotiation decision functions for autonomous agents,” *International Journal of Robotics and Autonomous Systems*, vol. 24, pp. 3–4, 1998.

[6] H. Raiffa, *The art and science of negotiation*. Cambridge, Mass: Belknap Press of Harvard University Press, 1982.

[7] Self-organizing ICT Resource Management (SORMA). [Online]. Available: http://www.sorma-project.eu

[8] M. Macías, O. Rana, G. Smith, J. Guitart, and J. Torres, “Maximizing revenue in Grid markets using an Economically Enhanced Resource Manager,” *Concurrency and Computation: Practice and Experience*, September 2008. [Online]. Available: http://dx.doi.org/10.1002/cpe.1370

[9] M. Macías, G. Smith, O. Rana, J. Guitart, and J. Torres, “Enforcing service level agreements using an economically enhanced resource manager,” in *Proceedings of hte 1st Workshop on Economic Models and Algorithms for Grid Systems (EMAGS 2007)*, Austin, Texas, USA, September 2007.

[10] S. Venugopal, X. Chu, and R. Buyya, “A negotiation mechanism for advance resource reservations using the alternate offers protocol,” in *Quality of Service (IWQoS 2008), 16th International Workshop on*, Enschede, The Netherlands, June 2008, pp. 40–49. [Online]. Available: http://dx.doi.org/10.1109/IWQOS.2008.10

[11] T. Püschel, N. Borissov, D. Neumann, M. Macías, J. Guitart, and J. Torres, “Extended resource management using client classification and economic enhancements,” in *Proceedings of eChallenges e-2007 Conference*, The Hague, Netherlands, October 2007, pp. 65–72.

[12] M. Macías and J. Guitart, “Influence of reputation in revenue of grid service providers,” in *Proceedings of the 2nd International Workshop on High Performance Grid Middleware (HiPerGRID 2008)*, Bucharest, Romania, November 2008, pp. 9–16.

[13] S. Angilella, S. Greco, F. Lamantia, and B. Matarazzo, “Assessing non-additive utility for multicriteria decision aid,” *European Journal of Operational Research*, vol. 158, pp. 734–744, November 2004.

[14] P. Barford and M. Crovella, “Generating representative web workloads for network and server performance evaluation,” in *SIGMETRICS ’98/PERFORMANCE ’98: Proceedings of the 1998 ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*, vol. 26, no. 1. New York, NY, USA: ACM Press, June 1998, pp. 151–160. [Online]. Available: http://dx.doi.org/10.1145/277851.277897

[15] J. Carroll, “The nature of the data, or how to choose a correlation coefficient,” *Psychometrika*, vol. 26, no. 4, pp. 347–372, December 1961. [Online]. Available: http://ideas.repec.org/a/spr/psycho/v26y1961i4p347-372.html