Maximising Revenue in Cloud Computing Markets by means of Economically Enhanced SLA Management

Mario Macías and Jordi Guitart

Technical University of Catalonia
Jordi Girona s/n
08034 Barcelona, Spain
{mario, jguitart}@ac.upc.edu

***Abstract*—This paper proposes a bidirectional communication between market brokers and resource managers in Cloud Computing Markets. This communication is implemented by means of an Economically Enhanced Resource Manager (EERM), that supports the negotiation process by deciding which tasks can be allocated or not, and under which economic and technical conditions. The EERM also uses the economic information that collects from market layers to manage the resources accordingly to concrete BLOs. This paper shows several Business Policies and Rules for maximizing the revenue of a Cloud Provider that sells its services and resources in a market. Their validity is demonstrated through several experiments that shown how the application of these rules can have a positive influence in the revenue and minimize the violations of Service-Level Agreements.**

I. Introduction

At recent years, the big mainframes paradigm in which users own their computing resources [1] is being progressively transiting to an utility-driven paradigm, in which users do not own resources and pay for the usage of remote resources [2]. Cloud Computing [3] is the most promising current implementation of Utility Computing in the business world, because it provides some key features over classic utility computing, such as elasticity to allow clients dynamically scale-up and scale-down the resources in execution time, or the possibility of customizing completely the software environment by acquiring administrator rights without putting in risk the whole system.

Since Clouds are heterogeneous, elastic and scalable, large systems are too complex to be managed centrally. Market-based resource management 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 to their real space and time requirements. Another advantage is that it is relatively easy to implement in a decentralised architecture, of which 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 a Cloud computing market, Brokers that represent Service Providers or Clients participate in a market to sell or buy services or resources. When the Clients find there their requirements, a negotiation process is started to establish the terms of the contract. If both parties reach an agreement, the terms of the contract are specified in a Service Level Agreement (SLA) and the Client 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 when they violate the SLA. Since negotiating Brokers are autonomous agents, it is needed to provide them with some business models and intelligent behaviour to allow them to be able to take the best decisions.

The SLA is composed by Service-Level Objectives (SLOs), which are shared between client and provider. The SLOs describe the terms of QoS that the provider is obliged to fulfil. This paper also uses the Business-Level Objective (BLO) as an internal parameter of the provider. The BLOs helps the provider to know whether its business goals are being achieved. The BLOs will determine the SLOs that the provider agrees, and the way it fulfils them.

This paper encompasses the autonomic enforcement of a single Business-Level Objective (BLO): the maximisation of the revenue. Achieving this objective is dulled by a limitation of current Cloud market models: the lack of communication between market and resource layers. Since the Resource Manager does not have knowledge about the business details, it is difficult to define accurate policies or objectives to be fulfilled by the resources. Similarly, if a Business Broker negotiates the sales of resources that it does not know, this would potentially lead to waste or overload resources. To deal with this problem, each Cloud provider requires a central entity that manages the resource allocations in order to satisfy the SLAs and maximize the metrics to fulfil its BLOs. This central entity is called Economically Enhanced Resource Manager (EERM), which must enable a bidirectional communication between Business and Resource Layers for taking good decisions in both Business and Resource management. Business brokers need resource-related information in order to adjust the business terms to

the real capacities of the Resource Fabrics. Resource Managers that survey for the correct fulfilment of the SLAs, require some business-related information in order to take the decisions that maximise the fulfilment of BLOs.

As in the real organisations, the BLOs or strategies of a Cloud Provider can change as times goes by. Reprogramming, recompiling and redeploying a new EERM each time the BLOs (or the way to satisfy them) change is not feasible in a production environment. The architecture of the EERM supports its customisation by means of rules that can be defined and dynamically tuned by the Cloud Administrator without need of stopping and redeploying a new EERM. This paper describes a rule-based EERM, of which rules can be specified by a Cloud Administrator for describing the reactive actions that the EERM must perform in some situations, such as negotiation requests from the market clients, resources overload, or task allocation requests.

The main goal of this paper is facing the problem of Cloud Resource Management from both business and performance points of view (often conflicting). The intention is providing an integral solution of several policies that work together to maximise the profit of a Cloud provider, dealing also with performance issues. Since the external conditions of the market and some internal parameters can change the absolute results, this paper also intends to evaluate the introduced policies in terms of relative results and tendencies (e.g. using certain policy you can generally increase the benefit). This paper describes and evaluates some rules for maximising the revenue: Dynamic Pricing, Resource Overprovisioning, Selective SLA Violation and cancellation, Ranges for QoS, Dynamic Reallocation of Tasks, and Redistribution of assigned resources.

The validity of policies are evaluated through simulations instead of real executions mainly because two reasons: the first is the difficulty to acquire a testbed large enough to get representative data from the experiments; the second is that this paper does not evaluate the performance, but the business benefits of using an EERM in Utility Computing markets. These markets are experimental at this moment, and there are not traces of real executions for their reproduction in the experiments.

The rest of the paper is structured as follows: after a summary of the related work, section III describes the scenario of the research, by defining some terms that will be used in the paper, the statement of the problem, and the proposed architecture to solve it. Section V describes the rules proposed for performing a BLO-oriented Resource Management. After section VI, in which the experiments and their results are described, the conclusions are summarized and some future work lines are explained.

II. Related work

There is a large body of work that considers Business-Oriented Resource Management. This section shows the works that are the most related with this paper. Yeo et al. [4] propose a model for market-based cluster computer with some elements common to EERM. The cluster nodes are connected to a central manager which incorporates other sub-components for performing Pricing, job scheduling, monitoring and dispatching, and so on. The main difference with EERM is that EERM is conceived to be integrated into a higher level grid market, so it does not implement some market functions such as accounting, billing, or identity provisioning. Freedman et al. [5] focus on Peer-to-Peer (P2P) content distribution by identifying explicitly highly demanded files and rewarding most those peers sharing highly demanded content. They use a market-based mechanism for allocating network resources across multiple files and play with the Law of Offer and Demand for incentivising providers to sell most scarce high-value resources (only files, but not CPUs or Memory). Their system is also designed so that the buy client chooses files consistent with its best interests, since it seeks to download at the current minimum price. This paper extends these policies for general purpose computing.

Pueschel and Neumann [6] use the concept of an EERM for optimising the revenue of a Cloud Manager. They apply policies as a heuristic and demonstrate their achievements on revenue maximisation or client classification when applying economic enhancements. This paper tries to be a step forward in the usage of policies, by allowing procedural policies that say to EERM what to do under certain conditions to get te best solution to given problems. This paper shows an upgrade of EERM that adds more policies, and performs a more intensive evaluation of their validity.

There are some other important works in Rule-Based Resource Management for distributed environments, such as the Collaborative Awareness Management by Herrero et al. [7] that promotes cooperation between resources for their optimisation by means of a set of rules, or the Business Rules Management System introduced by Schiefer et al. [8], which is able to sense and evaluate events in order to respond to changes in a business environment or customer needs. This paper tries to extend these approaches by combining High-Level Service and Business data with the low-level resource information, enforcing the flow of information between the two layers for their mutual optimisation.

Previous papers introduced some concrete policies, similar to the rules introduced in this paper. Sulistio et al. [9] proposed overbooking strategies for mitigating the effects of cancellations and no-shows for increasing the revenue. The overbooking policies implemented in this paper, in addition, considers the possibility of under-usage of the reserved resources from the client. Dube et al. [10] establishes different ranges of prices for the same resource and analyse an optimisation model for a small number of price classes. Their proposal is similar to the proposal of this paper of establishing Gold, Silver and Bronze ranges and optimising their QoS performance giving priority to the contracts that report the highest economic profit. This paper extends this work by combining the QoS ranges with several other policies, such as Pricing or Selective Violation.

Menasce et al. [11] demonstrated the importance of managing the resources having in mind the BLOs. They maximise

the revenue of e-Commerce servers by calculating dynamically Customer Behaviour Model Graphs and prioritising the workloads of the users that are willing to spend more money. Poggi et al. [12] introduce a similar approach, in which QoS for user sessions is shaped based on the commercial intention of the users. However, these models are not applicable to the scenario of this paper, since the Cloud Provider supports more generic types of workloads, not restricted to online shops, and does not interact with human customers, but with other client machines that automatically buy resources in a market.

Previous work from the authors demonstrated that Business-Oriented Resource Management could lead to an increase of the economic profit. That management could be in execution time [13], or in negotiation time [14]. This paper is intended to be a step forward from previous work by implementing and testing the validity of these business-oriented policies within an EERM with a rule engine [15] customizable by the system administrator, and by introducing new rules and policies, such as Selective SLA Violation or Resource Overprovisioning.

III. Resource Management to Satisfy BLOs

A. Previous definitions

Each EERM controls a set of physical machines. Each physical machine can host several Virtual Machines that can execute single tasks, such as Web Services or Batch Jobs.

All the tasks executed by a provider have an associated SLA composed by the next data: $SLA = \{Rev(vt), \overrightarrow{S}, \Delta t\}$, where:

- $\overrightarrow{S}$ are the Service Level Objectives (SLO) of the SLA, which specify the QoS terms of the job/service to be executed. By means of SLA decomposition [16], $\overrightarrow{S}$ must be translated to the low-level resource requirements, such as number of CPUs or amount of memory.
- Δt is the time period requested for allocating the task. It can be a fixed time period (for real-time tasks, such as Web Services), or a fixed-duration job that must be executed within a variable interval of time (used for batch jobs of which time requirements are not so strict).
- $Rev(vt)$ is the revenue function for fulfilling correctly or violating a SLA. In function of the time that the SLA is in violation status (expressed as vt), the revenue will decrease linearly until arriving to a maximum penalty. Let MP be the maximum penalty, MR the maximum revenue, MPT the maximum penalty threshold, and MRT the maximum revenue threshold, Equation 1 describes the function for calculating the revenue in function of the violation time vt (see Figure 1).

$$Rev(vt) = \frac{MP - MR}{MPT - MRT}(vt - MRT) + MR \quad (1)$$

B. Revenue Maximisation

In a distributed and heterogeneous organisation, all the entities have individual objectives, often in conflict with the objectives of other entities of the same organisation.

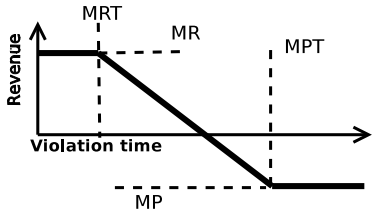

Figure 1. Revenue of a SLA in function of the violation time (Equation 1)

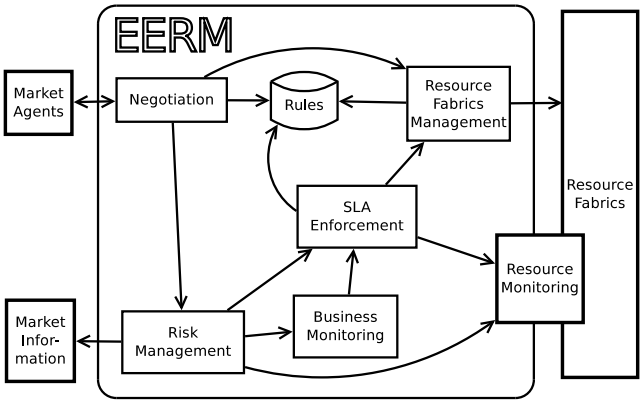

Figure 2. General architecture of the EERM

The objectives of a Cloud Provider must be defined by the Cloud Administrator. This paper is focused on Revenue Maximisation, which is the main motivation of most of the providers that enter in a Utility Computing Market: to earn money by selling their services.

This objective must be treated from two points of view:

Negotiation time: The EERM must help the market agents to achieve the most beneficial contracts for their BLOS. This paper implements policies for Dynamic Pricing and Overprovisioning of Resources that help to maximise the revenue in negotiation time.

Execution time: When the SLAs are agreed and the tasks are sent to the resources, the EERM must manage the resources for pursuing the optimum achievement of BLOs. Policies that are used in this paper at execution time are Selective SLA Violation and Cancellation, and Tasks Migration.

IV. Description of the Architecture

The problem raised in this paper is faced through an Economically Enhanced Resource Manager (EERM), of which architecture is depicted in Figure 2.

Negotiation: This component interacts with the clients in order to perform the resource allocation and pricing that fits best within its own objectives.

Risk Management: Taking into account both market and resources information, manages the risk for the EERM of doing determined actions, such as resource allocations, negotiation confirmations, etc. This risk is taken into account when negotiating and managing resources.

SLA Enforcement: Keeps track of tasks executed in the system and continuously watches the status of their SLAs. If SLAs are being violated it takes reactive measures for minimizing the economic and technical impact of the violations.

Resource Fabrics Management: It is the only component without any type of business knowledge. It acts as an interface to the resources pool, and orders the creation, destruction or modification of the Virtual Machines that handle the sold tasks. However, under certain events like the overload of the system, it can trigger certain rules that can describe some business-oriented actions.

Business Monitoring: It monitors the correct fulfilment of the BLOs. If it predicts that the objectives will not be achieved, it changes dynamically the Business Strategy and Rules. This component will be treated in future work.

V. Description of rules

As explained before, this paper only applies rules for achieving a single objective: the maximisation of the economic profit of a Cloud provider. For achieving the correct fulfilment of this BLO, it is required to provide the EERM with rules that describe what to do when some situations are triggered, such a resource failure, a SLA violation, a negotiation request from a client, etc. This section describes the rules used for maximising the revenue of Cloud providers.

A. Dynamic Pricing

In a competing market, provider must establish an adaptable pricing rule that varies in function of the offer/demand proportion, in order to gain the most clients as possible [17]. Previous work of the authors in this field established a formula that defined an aggressiveness factor as a function of the resources load status: the more loaded is the system, the more aggressive will be the negotiation. This is, the prices will be higher [14]. Having this in mind, our pricing rule consists in the maximisation of the next utility function:

$$u_{rv}(\overrightarrow{S}) = 0.5 + \frac{\sin\left(\frac{\pi}{2}\left(2u_p(\overrightarrow{S}) + \left(1 - a(\Delta t)^{15}\right)\right)\right)}{2} \quad (2)$$

Each of the components of utility function in Equation 2 is thoroughly explained in the previous article [14]. In this paper, however, there is a difference in the way the aggressiveness factor is calculated: instead of using current data for its calculation, future predictions about the resources load are used. Let $C_{used}(t)$ be a function that predicts the usage of the currently reserved resources over time in terms of CPU; let $C_{req}(t)$ be a constant function which represents the CPUs requested by the client in the negotiation process (as a result of the decomposition of the set of SLOs S); let $C_j(t)$ be a constant function that represents the number of CPUs of the physical resource j; Equation 3 shows how the aggressiveness factor $a(\Delta t)$ is calculated over a set of N physical machines.

$$a(\Delta t) = \frac{\sum_{j=1}^{N} \int_{t_i}^{t_f} C_{used}(t) + C_{req}(t)\, dt}{\sum_{j=1}^{N} \int_{t_i}^{t_f} C_i(t)\, dt} \quad (3)$$

Equation 3 is assuming that CPU is the bottleneck of the system. That is not always true, since in some other scenarios the critical resource could be the memory or the bandwidth. That should not be a problem for this model, since it supports any type of good. CPU is used having in mind some CPU-intensive services or batch tasks, just to exemplify the approach. The equations can be easily modified to allow the coexistence of several weighted goods.

B. Resource Overprovisioning

A classical RM will refuse an incoming reservation request from a client if there are not enough free resources for the requested time period. *Free resources* are those resources that are not reserved by any other client. However, these resources are often idle in concrete time periods (such as the early morning). If the RM allows to reserve resources that are already reserved by other clients but not used, the revenue could be noticeably increased.

When resources are overprovisioned for future tasks, it is needed an accurate predictor that is able to estimate the future usage of the resources with a small error rate in function to historical data. Based on this prediction, When a request arrives and there are not enough resources for allocating it, the scoring function in Equation 4 is calculated over the set of resources $j = \{1 \dots N\}$. Next, the physical resource j of which score is the higher positive one is selected as candidate for executing the incoming task and the pricing policy is triggered (see Section V-A). If there are not physical resources of which score is positive, the job is rejected.

$$score_j = 1 - \frac{\int_{t_i}^{t_f} C_{used}(t) + C_{req}(t)\, dt}{\int_{t_i}^{t_f} C_i(t)\, dt} \quad (4)$$

In this paper, the CPU is considered as the bottleneck for the most services to be executed in the Cloud Provider. However, Equation 4 can be extended to include other resources like memory, disk, or network bandwidth.

Indirectly, this function does not only perform resource overprovisioning, but also will allocate the batch jobs, negotiated with a variable time slot, at the off-peak hours, because these time slots will have better scoring as a consequence of the amount of idle resources.

C. Selective SLA Violation

Some situations, such as breakdowns in the resource pool or errors in overprovisioning estimations, can lead to a system overload that will not allow fulfilling all the SLAs. At this moment, the EERM can take measures for minimizing the economic impact of the SLA violations, such as renegotiation of tasks, migration of virtual machines to other idle hosts or, when none of the previous alternatives is possible, perform a selective violation of SLAs [13].

The selective SLA violation rule contains the criteria for choosing a set of Virtual Machines of which resources will be temporarily deallocated, in order to free enough resources so that the other Virtual Machines are able to fulfil their assigned

SLAs. For choosing the right Virtual Machines set, the next process is started: the set of machines of which resources must be temporarily deallocated is chosen by calculating the hypothetic total revenue if the violation time vt of the candidate SLA is updated, and all the revenues and penalties of the SLAs [13] are summated. After all the combinations are calculated, the SLA of which violation produces the higher gain (or the lower loss) is violated to leave free space for the other SLAs.

D. Selective SLA Cancellation

According to previous work [17], when a client sends a request for a task that cannot be executed by the provider because limitations of space, it is possible to cancel the tasks that are already scheduled or running in the system if the revenue of the incoming task is high enough to compensate for the penalty of the cancelled SLA.

This policy must be executed with caution, because the short-term benefit is in conflict with mid-term losses in the reputation of the provider [18]. However, this policy can be useful for minimize economic losses under certain conditions, such as system failures that produce massive SLA violations.

Because this paper does not consider reputation, a cancellation policy is applied without restrictions, for checking its effectiveness. When a client asks for a service, the policy of SLA cancellation is triggered: for each SLA of which time slot collides with the demand, and of which possible cancellation would free enough space to allocate the demand, it is calculated the possible benefit of cancelling it by subtracting the maximum penalty to the maximum price that can be asked to the client for (this is, the Reservation Price of the Buyer [19]). After all the SLAs have been checked, the SLA of which cancellation reports the highest benefit is marked as *cancellable* and the provider asks to the client its Reservation Price for allocating their tasks. If the provider accepts, the marked SLA is cancelled and the new task is allocated.

E. Ranges for Quality of Service

Given the aforementioned rules, it is possible to establish different ranges for QoS by tuning the values of MRT, MR, MPT and MP of the revenue function $Rev(vt)$ (see Section III-A).

In this paper, three types of QoS have been defined, from higher to lower: Gold, Silver and Bronze. The higher is the QoS range, the higher is MR and the lower is MP, and the nearer to 0 are MRT and MPT. The revenue function $Rev(vt)$ must be established in negotiation time as specified by the pricing policies. When the system is overloaded and some SLAs in the system are in danger of violation, the definition of different ranges of QoS combined by the selective SLA violation and cancellation rules makes the EERM to violate preferably SLAs with low QoS.

F. Tasks Reallocation

Due to the heterogeneous nature of the Cloud, it is possible that the system becomes unbalanced and some resources in the pool are overloaded. To deal with this issue, when the load of a resource is over 90%, the EERM triggers a rule that migrates some tasks to other resource with enough free space until the load of the resource is lower than the 90% threshold. In addition, this policy could also be used when a SLA is violated: the EERM tries to migrate its associated task, or migrate other tasks to leave enough free space.

Recent studies [20] reveal that the cost of migrating web services in Cloud Computing is near zero thanks to virtualization, because creating, booting, and populating a virtual machine with data takes few seconds (negligible in tasks that take from one to several hours). This policy could also be used for Energy-Efficiency policies, by means of workload consolidation [21].

G. Redistribution of Assigned Resources

In addition to previous policy, this policy takes the most of Cloud Computing advantages over previous Utility Computing paradigms. The sales of services are negotiated in terms of QoS, for example throughput, response time, video quality, etc. However, when the tasks are allocated, a concrete amount of plain resources are assigned to them, such as CPU, Memory, Network Bandwidth, etc. In negotiation and allocation time, the EERM performs a SLA Decomposition process [16] for allocating the correct number of plain resources for a task.

The SLA Decomposition, however, is not an exact process: errors can occur when translating from High-Level QoS terms to plain resources or vice-versa. In addition, these errors can derive to a violation of a SLA, if too few resources are allocated for a service, or to a waste of resources, if the EERM allocates more resources than needed.

This paper proposes to compensate the inaccuracies of SLA decomposition by performing redistribution of resources: when a SLA is being violated because it has not enough resources allocated and there are not idle resources in the Hardware Machine for assigning them to the task, the EERM will look for the tasks that are underutilizing their resources. If there are enough underutilized resources, a sufficient portion of them will be unassigned to their current tasks, and assigned to the task with insufficient resources. Virtualization technology makes it easy to implement this policy [22].

VI. Evaluation

A. Experimental Environment

Two actors participate in the market: Clients and Resource Providers. The resources are offered by the providers, formed by an EERM that manages and controls a pool of hardware resources. Depending of its kind of workload, a Web Client or a Grid Client enters in the Market to respectively ask for plain resources to host a web server or to run a job. The workload for Grid has a pseudo-random distribution and the workload for Web services follows a typical distribution, taken from a real Web application, with a variable workload in function of the hour of the day [23].

Clients send a SLA proposal that specifies the plain resources to buy, the duration of the job, and a time interval

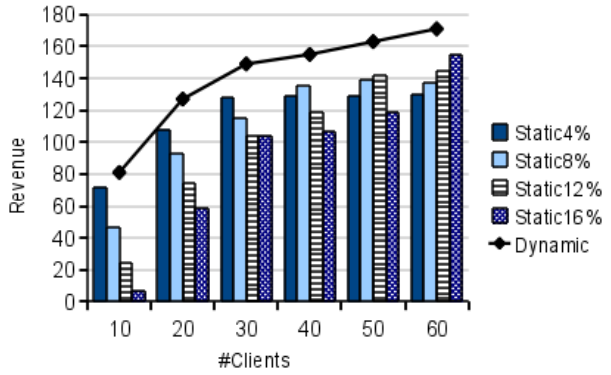

Figure 3. Comparison of revenue between dynamic and fixed pricing

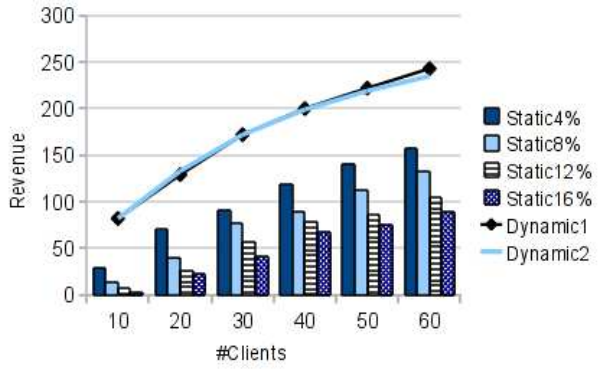

Figure 4. Comparison of revenue between dynamic and fixed pricing when two dynamic providers are in the market

when the job can be executed (bigger than the duration, so the EERM schedules the best time slot). Web 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 at that given moment).

Both Batch and Web Clients must specify what QoS class they want: Gold, Silver, or Bronze. For the same task in equal time and load conditions, Gold QoS tasks have a Reservation Price for the seller 50% higher than the Bronze QoS Reservation Price, and Bronze QoS tasks have a Reservation Price 20% higher than the Bronze QoS one. The price is then established as explained in Section V-A.

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 the Client a time allocation and a price, as decided by following their rules. Finally, the Client chooses the Provider with a best price and time schedule for its interests and sends him a confirmation.

The Provider can violate a SLA due to a bad prediction of the expected workload when performing resources overprovisioning or SLA decomposition.

B. Dynamic pricing

Five providers with 12 CPUs each one are competing in a market of which clients send Bronze, Silver and Gold tasks at the same proportion. There is one provider that implements dynamic pricing, and four providers that offer fixed prices, which are always a fixed proportion between the Reservation Price of the Seller and the Reservation Price of the Buyer.

Figure 3 shows the revenues of the dynamic pricing provider and providers with a fixed price that ranges from the 4% to the 14% over the Reservation Price of the Seller (labelled from *Static4%* to *Static16%*). Paying attention only to the fixed-pricing providers, it can be seen that the provider with lowest fixed prices (4%) get the highest revenue in the offer excess scenario (only 10 clients for 5 providers), the provider with highest fixed prices (16%) get the highest revenue in the demand excess scenario (60 clients for only 5 providers), and the providers with average prices get the highest revenues in equilibrium scenarios. Comparing them with the Dynamic Pricing provider, it can be seen how it beats all the other providers in all the scenarios due to its capacity of adaptation.

In order to check the effects of competence between several dynamic-pricing providers, the same experiment has been repeated, by adding a new provider with dynamic pricing policies. Figure 4 shows that the benefit of most fixed-pricing providers decreased noticeably in all the scenarios. The effects of new competence pushed the providers to keep the prices low, and only *Static4%* provider has been able to keep similar benefits in both scenarios.

Somebody could think that the addition of a new provider, by keeping the same number of clients, would cause a general decreasement in benefit of all providers. However, figure 4 shows that the dynamic-pricing providers (labelled as *Dynamic1* and *Dynamic2*) have now more benefits than in the experiment of figure 3. The strong is the competence, the higher share of market acquire dynamic-pricing providers. Both dynamic-pricing providers have very similar economic benefit because their best adaptation to market dynamics.

C. Resources overprovisioning

For comparing the effectiveness of Resource Overprovisioning policies for Revenue Maximisation, two providers are competing in a Market. The first provider performs Dynamic Pricing and Overprovisioning, and the second only performs Dynamic Pricing. Both providers manage two 8-CPU servers that will host the Virtual Machines where the tasks will be executed. The experiments are repeated with several demand amounts: from 5 to 35 simultaneous clients that will ask periodically for Virtual Machines for hosting their services, repeating this process during one week. The size of the virtual machines can vary from 1 to 4 CPUs, in function of the hour of day (1 CPU in off-peak hours, 4 CPU in peak hours).

The predictor component, which decides if an incoming task can be allocated or not based on predictions of workload, has an error rate of 10% that, according to current work in the field, it is a reasonable rate [16].

Figure 5 compares the revenue of the two providers. It shows how the revenue of a provider stops increasing linearly with the number of clients, because its resources tend to be al-

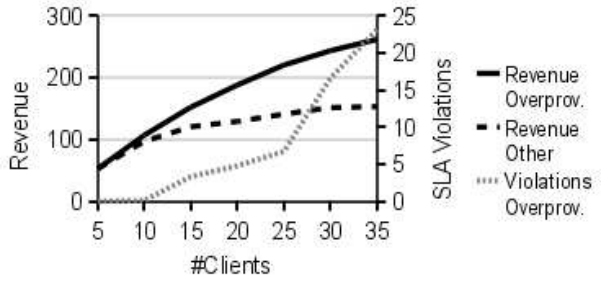

Figure 5. Comparison of revenue and SLA violations with and without Resource Overprovisioning

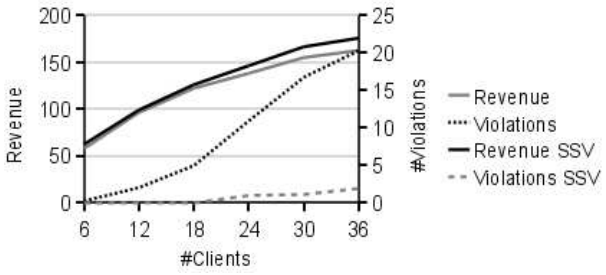

Figure 6. Providers with selective SLA violation (SSV) earn more money than providers without it

most full and the difficulty of finding free slots tend to infinite. However, a resource with overprovisioning can allocate more tasks in certain time slots, so its revenue continues increasing with clients more than the provider without overprovisioning.

The drawback of resource overprovisioning is the possibility of violating SLAs. It is important to define what is accounted as violation by the EERM. A SLA can be fulfilled completely (accounted as 0 violations), completely violated (accounted as 1 violation) or partially violated (accounted with a value between 0 and 1). If the time at which the provider is not giving to the client the agreed QoS is lesser than MRT (see Figure 1) it is considered that there are no violations. If the duration of the violation time is higher than MPT it is considered a complete violation. For a partial violation, its value for accounting is calculated as $\frac{vt-MRT}{MPT-MRT}$.

Figure 5 shows that the number of violations is increased exponentially when clients are increased linearly. The ~ 20 violated SLAs in the simulation with 35 clients represents the $\sim 1\%$ of the ~ 1800 total agreed SLAs.

Next section shows how these violations can be minimized by selecting which SLAs must be violated when the system becomes overloaded.

D. Combining Ranges for Quality of Service and Selective SLA Violation

In this experiment, two providers with 8 CPUs are competing in a market. Both perform Dynamic Pricing and Overprovisioning, but only one implements Selective SLA Violation. The experiments are repeated with a variable number of clients (from 6 to 36) that ask for different ranges of QoS (Gold, Silver, and Bronze).

Figure 6 shows how the provider that implements Selective SLA Violation earns between 5-10% more money than providers without it. The most important result is that with Selective SLA Violation, the violation of SLAs decreases $\sim$90%, because the EERM focuses the violations in those SLAs that still have margin of violation before the violation time surpasses MRT, as explained in previous section.

The difference between the QoS ranges is how MRT, MR, MPT and MP are located. The experiments of this paper use the next values: $MRT(Bronze, Silver, Gold) = (15\%, 5\%, 3\%)$, $MPT(Bronze, Silver, Gold) = (75\%, 50\%, 30\%)$ and

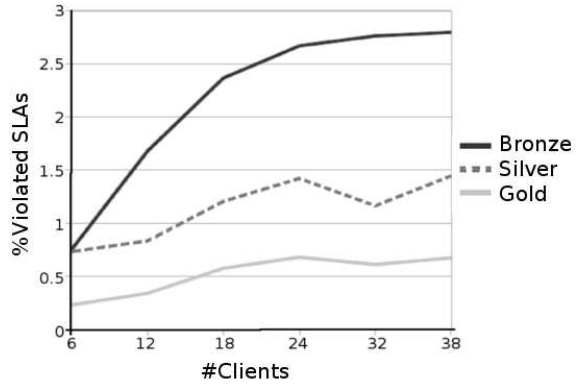

Figure 7. Different ranges of QoS for SLAs have different failure rates

$MP(Bronze, Silver, Gold) = (MR, 2MR, 3MR)$. These values are arbitrary, but they reflect what they must do: Gold clients are less allowed to have violations than Silver, and Silver less than Bronze, because the penalty pitch happens earlier, is more sloping, and the economic penalty is much higher. The value of MR is negotiated with the clients. Changing these values would not alter qualitatively the simulation results. Figure 7 shows how the selective SLA Violation behaves with different ranges of QoS.

E. Tasks Reallocation

In this experiment, two providers with 4 physical machines of 8 CPUs each one are competing in a market with all the policies tested previously in this section, plus a Tasks Reallocation policy in one of the providers. Figure 8 shows how reallocating dynamically tasks when the system surpasses a load threshold (80% in the experiments), increaes the revenue up to 12% and minimises significantly the violation of SLAs in scenarios with a high excess of demand.

F. Selective SLA cancellation

In this scenario, two providers with a single machine of 8 CPUs each one are competing in a market. Both providers are configured to perform Dynamic Pricing, Resources Overprovisioning and Selective SLA Violations. Only one performs Selective SLA Cancellations. The same experiment is repeated with different number of clients.

Figure 9 shows how the revenue is greatly increased by applying Selective SLA cancellation ($\sim$50% over the non-

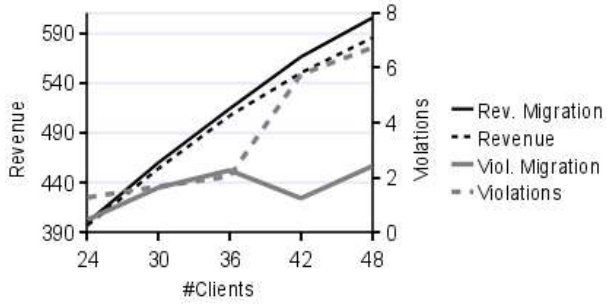

Figure 8. Providers that implement service migrations minimise SLA violations and increase their revenue

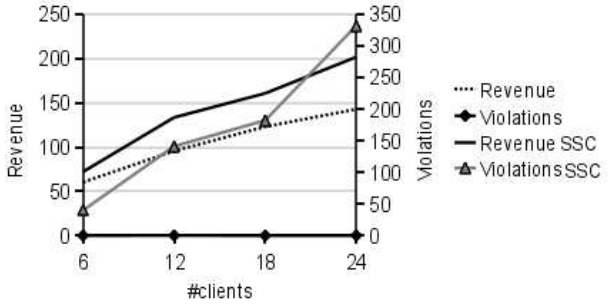

Figure 9. Selective SLA cancellation increases greatly the revenue, but also the violations of SLAs

cancelling provider), but also the violation of SLAs (in the 24-client scenario is 35000% over the non-cancelling provider). In a real scenario that implements a reputation system this policy would not be valid. However, this policy could be applied in certain situations, such as arrival of tasks from a special user, reorganisation after partial failures of the system, etc.

G. Softening peak-hours

As explained at the end of Section V-B, the scoring function of Equation 4 will indirectly allocate batch tasks that can be allocated during a variable time range, in time slots of which demand is low. To illustrate this, the next experiment is performed: two servers are accepting jobs, having activated the rules of dynamic pricing, resource overprovisioning and selective SLA violation. The only difference is that one of the servers has disabled the scoring function of dynamic pricing when it is trying to allocate the best time slot for an incoming batch task, and it allocates it in the first time slot with enough free resources. In the market, there are 12 Web Services and 12 Batch Jobs clients that are sending their requests periodically to the market. The proportion of Gold, Silver and Bronze QoS are the same within the tasks.

Figure 10 shows the amount of money that each provider earns in each minute of the day. The curves are slightly less sharpened in the provider that applies softening of peak hours. It earns less money in peak hours, but during off-peak hours (the early morning) it earns more money. Since the provider that performs peaks minimization is not so overloaded in peaks of demand, there is less possibility of violations.

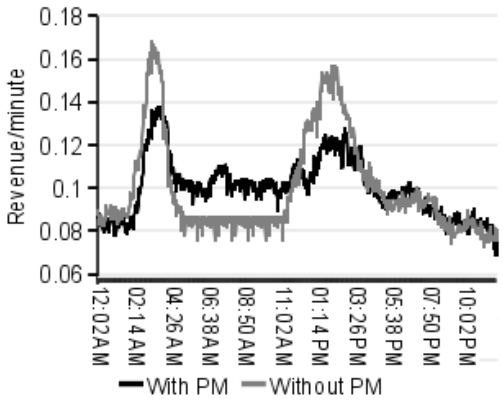

Figure 10. Comparison of the revenue of each minute between a provider with peak minimisation and other provider without it

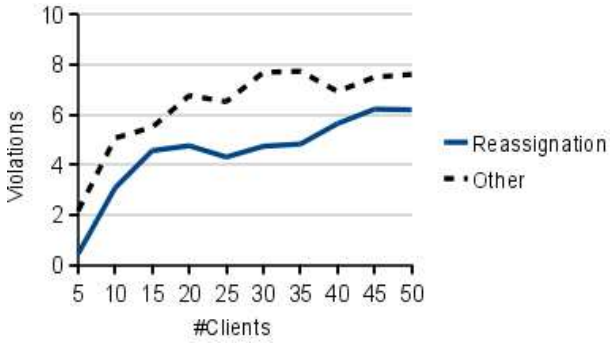

Figure 11. Comparison of SLA violations between providers with and without redistribution of assigned resources

Another probability is, as shown in previous work [14], that the provider with a peak softening can offer best prices for batch jobs, so it will allocate more tasks and increase its revenue.

H. Redistribution of allocated resources

In this scenario, two providers with 16 CPU are competing in a services market. Both providers implement dynamic pricing and resource overprovisioning, but only one implements dynamic resource redistribution.

The violations of SLAs caused by inaccuracies in decomposition have been scored separately to show only the influence of resource redistribution, without interferences of violations due to other causes. Since this policy does not distinguish between Gold, Silver, and Bronze tasks, the QoS ranges are uniform for not altering the results with some random noises due to the QoS range of the task of which resources are redistributed.

Since errors in decomposition have a random nature, the experiments have been repeated several times and the average values of SLA violations are shown in figure 11. It shows the difference of violations is not enough high to have an impact in the revenue (and that is why revenue is not shown), but in systems that implement reputation, these differences in the number of violations

VII. Conclusions and Future Work

This paper proposed an architecture for establishing a bidirectional communication between market brokering and resource management tasks in Cloud Markets. Its implementation, the Economically Enhanced Resource Manager, supports the negotiation process by deciding which tasks can be allocated or not, and under which economic and technical conditions; the EERM also uses the economic information that collects from market layers for managing the resources accordingly to concrete BLOs.

This paper introduces several Business Policies and Rules for pursuing a concrete Objective: the maximisation of the revenue in a market-based Cloud Provider: Dynamic Pricing, Resources Overprovisioning, Tasks Reallocation, different ranges for QoS, Selective SLA violation and cancellation, Minimisation of Peak-hours, and Redistribution of Resources. Their validity has been demonstrated through several experiments that shown how the use of these rules can have a positive influence in both the short-term revenue and the long-term, by minimising the SLA violations, which lead to maximising the reputation, and the possibility of offering higher prices in future negotiations [18].

The work of this paper will be continued in two research lines: the first one is the creation and evaluation of similar policies, but applied to other BLOs, such as Client Classification. The second research line is to add support for dynamic rules that are continuously being automatically modified by the EERM for allowing a better adaptation of the policies to a changing environment such a market.

Acknowledgements

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.

References

[1] J. Basney and M. Livny, "Deploying a high throughput computing cluster," in *High Performance Cluster Computing: Architectures and Systems, Volume 1*, R. Buyya, Ed. Prentice Hall PTR, 1999.

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

[3] R. Buyya, C. S. Yeo, and S. Venugopal, "Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities," *High Performance Computing and Communications, 10th IEEE International Conference on*, vol. 0, pp. 5–13, 2008.

[4] 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.

[5] M. J. Freedman, C. Aperjis, and R. Johari, "Prices are right: Managing resources and incentives in peer-assisted content distribution," in *Proc. 7th International Workshop on Peer-to-Peer Systems (IPTPS08)*, Tampa Bay, FL, Feb. 2008.

[6] T. Pueschel and D. Neumann, "Management of cloud infrastructures: Policy-based revenue optimization," in *International Conference on Information Systems (ICIS 2009)*, Phoenix, Arizona, December 2009.

[7] P. Herrero, J. L. Bosque, M. Salvadores, and M. S. Perez, "A rule based resources management for collaborative grid environments," *Int. J. Internet Protoc. Technol.*, vol. 3, no. 1, pp. 35–45, 2008.

[8] J. Schiefer, S. Rozsnyai, C. Rauscher, and G. Saurer, "Event-driven rules for sensing and responding to business situations," in *Inaugural International Conference on Distributed Event-Based Systems (DEBS 07)*. New York, NY, USA: ACM, 2007, pp. 198–205.

[9] A. Sulistio, K. H. Kim, and R. Buyya, "Managing cancellations and no-shows of reservations with overbooking to increase resource revenue," *Cluster Computing and the Grid, IEEE International Symposium on*, vol. 0, pp. 267–276, 2008.

[10] P. Dube, Y. Hayel, and L. Wynter, "Yield management for it resources on demand: analysis and validation of a new paradigm for managing computing centres," *Journal of Revenue and Pricing Management*, vol. 4:1, pp. 24–38, 2005.

[11] D. A. Menascé, V. A. F. Almeida, R. Fonseca, and M. A. Mendes, "Business-oriented resource management policies for e-commerce servers," *Perform. Eval.*, vol. 42, no. 2-3, pp. 223–239, 2000.

[12] N. Poggi, T. Moreno, J. Berral, R. Gavalda, and J. Torres, "Self-adaptive utility-based web session management," *Computing Networks*, vol. 53:10, pp. 1712–1721, 2009.

[13] 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*, p. n/a, September 2008. [Online]. Available: http://dx.doi.org/10.1002/cpe.1370

[14] M. Macías and J. Guitart, "Using resource-level information into nonadditive negotiation models for cloud market environments," in *12th IEEE/IFIP Network Operations and Management Symposium (NOMS'10)*, Osaka, Japan, April 2010, pp. 325–332.

[15] "Drools expert." [Online]. Available: http://www.jboss.org/drools/drools-expert.html

[16] G. Reig, J. Alonso, and J. Guitart, "Prediction of job resource requirements for deadline schedulers to manage high-level slas on the cloud," in *9th IEEE International Symposium on Network Computing and Applications (NCA'10)*.

[17] 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.

[18] M. Macías and J. Guitart, "Influence of reputation in revenue of grid service providers," in *2nd International Workshop on High Performance Grid Middleware (HiPerGRID 2008)*, Bucharest, Romania, November 2008.

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

[20] I. Goiri, F. Julia, and J. Guitart, "Efficient data management support for virtualized service providers," in *17th Euromicro Conference on Parallel, Distributed and Network-based Processing (PDP'09)*, Weimar, Germany, February 2009, pp. 409–413.

[21] J. Torres, D. Carrera, V. Beltran, N. Poggi, K. Hogan, J. Berral, R. Gavald, E. Ayguad, T. Moreno, and J. Guitart, "Tailoring resources: The energy efficient consolidation strategy goes beyond virtualization," in *5th IEEE International Conference on Autonomic Computing (ICAC 2008)*, Chicago, Illinois, USA, June 2008, pp. 197–198.

[22] I. Goiri, F. Julia, J. Ejarque, M. de Palol, R. Badia, J. Guitart, and J. Torres, "Introducing virtual execution environments for application lifecycle management and sla-driven resource distribution within service providers," in *8th IEEE International Symposium on Network Computing and Applications (NCA'09)*, Cambridge, Massachusetts, USA, July 2009, pp. 211–218.

[23] P. Barford and M. Crovella, "Generating representative web workloads for network and server performance evaluation," in *1998 ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems (SIGMETRICS '98/PERFORMANCE '98)*, vol. 26, no. 1. New York, NY, USA: ACM Press, June 1998, pp. 151–160.