A Procurement Market to Allocate Cloud Providers' Residual Computing Capacity

Paolo Bonacquisto, Giuseppe Modica, Giuseppe Petralia, Orazio Tomarchio

A procurement market to allocate cloud providers' residual computing capacity

Paolo Bonacquisto, Giuseppe Di Modica, Giuseppe Petralia, Orazio Tomarchio

Department of Electrical, Electronic and Computer Engineering
University of Catania
Catania, Italy
firstname.lastname@dieei.unict.it

Abstract. Commercial cloud providers are used to allocate computing resources to requesting customers according to the well known direct-sell, fixed-price mechanism. This mechanism is proved to be economically inefficient, as it does not account for the market's supply-demand rate. Nevertheless, providers will unlikely abandon a pricing mechanism which is very easy and cheap to implement in favour of alternative schemes. On the other end, none of the commercial providers adopting the fixed-price mechanism is able to allocate their overall computing capacity. Not selling a single virtual machine within a predefined time slot means a profit loss to the provider. Alternative mechanisms are therefore needed to sell what we call the "residual" computing capacity, i.e., the capacity which the provider is not able to allocate through direct-sell. We argue that auction-based sells may meet this need. In this paper the design of a procurement market for computing resources is proposed. Also, an adaptive bidding strategy has been devised to help providers to maximize the revenue in the context of procurement auctions. Simulations have been run to test the responsiveness of the strategy to the provider's business objective.

1 Introduction

Cloud computing has emerged as a key technology for the realization of scalable on-demand computing infrastructures, where resources are provided to remote customers on the basis of Service Level Agreements (SLAs). Several features, such as virtualization of hardware, scalability, elasticity enable Clouds to adapt resource provisioning to the dynamic demands of Internet users. Resources are thus provided to customers as other public utilities like water or electricity, following a commodity market model [3].

In such a market commercial providers compete to offer their services, while customers compete to acquire resources on the basis of their Quality of Service (QoS) and pricing requirements [8]. Given the high dynamism of resources' availability and workloads, meeting the QoS constraints and maintaining an acceptable level of system performance and utilization are some of the primary problems to tackle. Effective resource allocation strategies must be devised that,

besides the technical features, also take into account the business features. We claim that benefits for both the customers and the providers may be obtained from adopting resource allocation strategies based on market principles.

One factor that strongly influences a market-based resource allocation scheme is the **pricing model**. The pricing strategy mostly adopted by main commercial IaaS providers to allocate virtual machines to requesting customers is known as "On-Demand". According to this strategy, customers are charged for the time frame during which the resource is actually utilized[1]. Providers ask customers to pay a fixed price for accessing computing capacity by the hour. Though the "fixed-price" scheme is considered to be economically inefficient, it is easily applicable to the cloud paradigm. There is apparently no evident reason for providers to abandon the direct-sell, fixed-price scheme in favor of alternative schemes. Especially for long-term requests, direct-sell is profitable to providers.

It is a matter of fact, however, that providers are not able to allocate their full computing capacity. Taking a look at the one-hour time window, there is a variable portion of computing resources (which we are going to name *spare* resources) which remain unsold and therefore do not produce income. We argue that if direct-sell fails to allocate 100% of providers' nominal computing capacity, alternative (possibly supply-demand based) pricing schemes should be adopted to allocate the spare capacity. Provided that costs for running the spare machines are covered, providers may be willing to sell that capacity at lower prices. So, on the one end providers may be interested in allocating the spare capacity to short-term customers' requests at a supply-demand regulated price (for longer commitments the regular direct-sell is more convenient). On the other end customers needing computing capacity for very short periods might want to obtain it at (lower) market prices.

In this paper, we propose to employ a **procurement auction** mechanism to allocate spare computing resources. We analyze the factors that mainly impact the strategic choices of providers in the acquisition of the goods allocated through auctions. The purpose of this work is to define a bidding strategy which guides the providers in the choice of the right actions to take in the context of a procurement process in order to maximize their business objective. In the addressed market scenario, our attention is devoted to the optimization of the utilization rate of providers' data centers. The remainder of the paper is structured as follows. Section 2 proposes a review of the literature and discusses the rationale of the work. Section 3 introduces the proposed idea and delves into technical details about the procurement auctions. Section 4 describes the proposed adaptive strategy to be used by the provider when participating in procurement auctions. In Section 5 simulation results are presented and discussed. Finally, the work is concluded in Section 6.

[1] http://aws.amazon.com/ec2/,
http://www.microsoft.com/windowsazure/,
http://www.rackspace.com/

2 Motivation and literature review

Many IT researchers are very much concerned with the application of auction mechanisms to the problem of optimal allocation of computing resources [13, 5]. For the majority of researchers, combinatorial auctions are the most appropriate sale mechanism for allocating virtual machines in the cloud. In combinatorial auctions the participants bid for bundles of items rather than individual items [6]. This mechanism seems to perfectly fit the Cloud context, as customers usually need to acquire not just one resource but a bunch of resources. In [17] authors address the scenario of multiple resource procurement in the realm of cloud computing. In the observed context, they pre-process the user requests, analyze the auction and declare a set of vendors bidding for the auction as winners based on the Combinatorial Auction Branch on Bids (CABOB) model. In [14] a combinatorial, double-auction, resource allocation model is instead proposed. The efficiency of the proposed economic model is proved in the paper, but to our advice that idea is not technically viable since a bundle allocated to a customer is composed of computing resources offered by different providers, thus forcing the customer to deploy their application on a geographically distributed cluster of machines.

The auction mechanisms proposed so far in the literature put the provider in a privileged position in the market: computing resources are seen as scarce and precious goods, whose allocation is carried out through competitions run among customers. We argue this viewpoint must be overturned. Spare resources are resources which providers do not manage to allocate through direct-sell. From the provider's perspective they must be regarded as perishable goods that need to be sold within a certain time frame otherwise they get wasted. Not selling a virtual machine in a given time slot means a profit loss to the provider, who is spending money anyway to keep the physical machines up and running. We then look at the trade of computing resources from a new perspective, in which providers, in the aim of maximizing their data center's utilization, may be willing to attract customers by lowering the offer price. On their turn, customers may get what they need, at the time they need it, at a price which is lower than the standard price at which they usually buy.

In the last few years, Amazon has been trying to allocate its spare resources through the *Spot Instance* model[2]. This model enables the customer to bid for what they call unused computing capacity. Though this model represents the very first attempt to build up a virtual market of computing resources regulated by market prices, it is still unclear and is not proved to be resistant to potential malicious behaviors of customers (dishonest customers can abuse the system and obtain short-term advantages by bidding large maximum price bid while being charged only at the lower spot price [18]). Furthermore in [1] authors prove that the Amazon's Spot Price is not market driven, rather is typically generated as a random value near to the hidden reserve price within a tight price interval.

[2] https://aws.amazon.com/ec2/purchasing-options/spot-instances/

We advocate that the market model best fitting the just described perspective is the one which provides for the sale of computing resources through **procurement auctions**. Procurement auctions [10] (also called *reverse auctions*) reverse the roles of sellers and buyers, in the sense that the bidders are those who have interest in selling a good (the providers), and therefore the competition for acquiring the right-to-sell the good is run among providers.

Smeltzer et al. [15] outlines potential advantages and drawbacks of adopting reverse auctions for goods allocation. Further, they point out the appropriate conditions which must apply for the reverse auction to be effective and convenient to both goods' suppliers and buyers. The most important are a clear specification of the commodity to be allocated and the fragmentation of the market. As for the first point, in the cloud community there is a common understanding of computing capacity's technical specification: information such as core numbers, CPU speed, RAM size, etc. are the only data needed to clearly and unequivocally state the product specification. With respect to the second point, we are proposing an open market of computing capacity where customers may look for spare resources to buy at lower prices, and that will naturally attract many providers interested in allocating spare resources and increasing their market share. Clear advantages for providers are that they may find customers in one single big market with no extra effort and that the market gives them the chance to maximize the occupancy rate of their data centers; on the other side, customers do not have to search for providers' offers and will get the requested computing capacity at a lower, supply-demand regulated price.

3 The procurement process

In this section we discuss the design of an open market of computing capacity to which any provider and any customer is admitted, and where computing resources can be sold through auction-based allocation schemes. The perspective is that of *procurement auctions*, where an initial price is called out on a good/service, and bidders iteratively have to call lower prices in order to gain the *right-to-serve*. The market mechanism is the following. Customers communicate their computing demand to the market. A *broker* will take care of demands. For each specific demand, the broker (auctioneer) will run a public auction in which any provider (bidder) can participate and compete for acquiring the right to serve the demand. The winning provider (who offered the lowest price) will eventually have to serve the customer's demand. Being the auctions open to the participation of multiple providers, the competition is granted. Providers will have to fight to gain the right-to-serve the demand. For a given demand, the determination of the final price is driven only by the evaluation that each provider has on the demand to be served. Advantages for customers are clear: they will get their demand served at the lowest price. Further, they will no longer have the burden to search for providers, as providers gather autonomously in the market.

Focus in this paper is on two different types of procurement auctions. The common part of the two auction mechanisms is the preparation phase: it provides

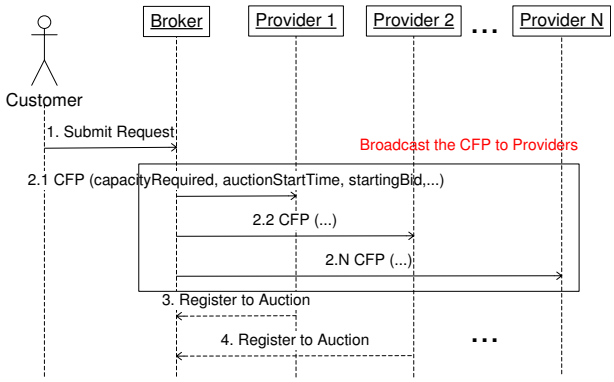


Fig. 1. Auction process: distribution of the CFP

that upon the arrival of a demand, the broker issues a public "call for proposal" (CFP) to invite providers. The CFP shall specify a minimum set of auction parameters including the start-provision time, the stop-provision time, the initial price (from which discount bids are expected), the bidding rules (who can bid and when, restrictions on bids) and the clearing policy (when to "terminate" the auction, who gets what, which price has to be paid). Figure 1 depicts the involved actors and the messages exchanged to carry out the auction preparation phase.

After collecting the willingness of providers to participate in the auction, the preparation phase ends up and the competition starts according to what is specified in the CFP. Basically, the broker will launch a number of competition rounds which depends on the type of auction advertised in the CFP. When the exit condition specified in the clearing policy holds true, the winner is appointed and is communicated to all participants along with the final price. Figure 2 depicts the just described steps.

What makes one auction mechanism different from another is the information specified in the bidding rules and the clearing policy respectively. For our purpose, in this paper the following auction types will be addressed: English Reverse (ER) and Second Price Sealed Bid (SPSB) [12]. The ER is a multi-round auction. The CFP specifies the initial price from which discounting bids (offers) are expected. The participating bidders may post their offers. Discounting offers are called out, so that every bidder is always aware of the reference price for which further discounts are to be proposed. If no offer arrives within a timeframe (publicly set in the CFP), the good will be assigned to the last best (i.e., the lowest priced) offer. This type of auction allows bidders to gather information of each other's evaluation of the good. The SPSB is a single round auction. All bidders have the chance to bid just once before the auction is cleared. When bidders receive the CFP, they check the initial price and decide to either bid or not to bid. After all participants have posted their bid, the broker clears the auction and allocates the "demand" to the second best bidder. The peculiarity of this auction is that bidders are not aware of each other's offer (only the win-

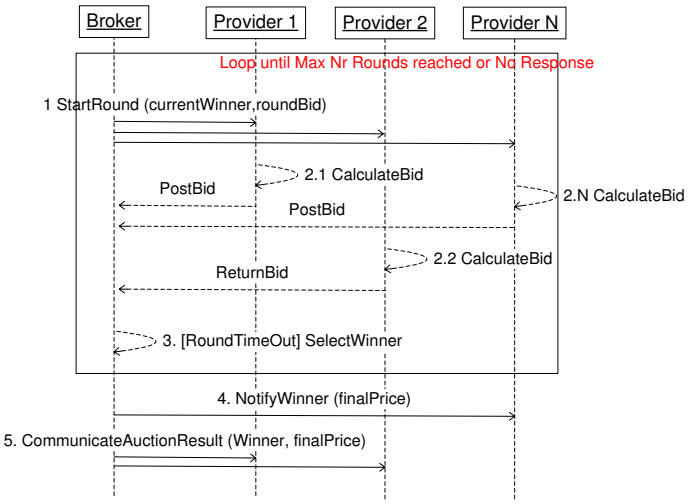


Fig. 2. Auction process: winner determination

ning bid will be broadcast at the end of the auction) and that the winner will be acknowledged a price which is higher than their own bid, thus increasing their overall utility.

As a basic market rule, a provider is admitted to participate in any CFP they like, with the obligation that if they win the auction, they are committed to serve the customer's demand, otherwise they will incur a penalty. The commitment rule mandates the provider to reserve resources for the CFPs they take part in. Those resources will remain "locked" until the auction is cleared, i.e., the resources may not be considered available to accommodate any new demand or to participate in a new auction. In a few words, the participation of a provider to an auction is subjected to the provider's availability of the amount of resources for which the auction has been called. If we consider that, on a statistical base, a provider will unlikely win the 100% of the auctions they take part to, there will always be an amount of resources which will remain not utilized because they are locked (i.e., awaiting for the respective auction to be cleared). In some previous works [7, 2] we touched on the phenomenon of underutilization of data centers in procurement-based markets. In this work we will address that problem and propose to overcome it by applying the mechanism of **resource overbooking** [16] to the cloud market. In particular, our objective is to investigate the impact of such a mechanism on the utilization level of data centers. Section 5 provides interesting feedbacks on that study. The overbooking lets providers compete for more customers' demands than they are able to eventually serve. Providers may participate in a given auction even though, at the time of joining the auction, they have no resource available to serve the demand which is object of that auction. But, what happens if a provider is out of resources at the time they are appointed the auction's winner? If we want this mechanism to be fair and transparent to customers, some market rules must be enforced that grant the

customer's right to receive the service on the one hand, and penalize providers that overuse the overbooking on the other one. One of the objectives of this work is to define the "reassignment" policy, i.e., the actions to be taken to grant the delivery of the service to the customer in the case that the provider(s) appointed as winner(s) can not provide it. This policy must state a) who among the remaining participants is assigned the right-to-serve, b) who is in charge of paying the penalty and c) how much is due. In this respect some proposals can be found in the literature. In [5] authors discuss some penalty functions which may be used in what they call "computational economies". Those functions may be classified into constant and dynamic ones. The former provide for an application of a constant penalty for those who win the auction but are not able to serve the demand (*defaulting providers*), the latter apply different penalties according to the impact of violation made by each party. We designed a reassignment scheme which tries to re-assign the right-to-serve among those who participate in the auction and applies dynamic penalties. The general rule is that if a winning provider is defaulting, the right-to-serve is passed to the next best-offering provider. This process may iterate until a provider is found which is able to serve the demand. In the case that all participants are defaulting, the CFP is re-issued (every defaulting participant is given a constant penalty). The general penalty rule is that every defaulting provider in the chain will have to pay a fee that is proportional to the difference between the final price (that called by the provider who will eventually serve the demand) and the price they had called out. The proportionality is implemented through the concept of price "distance" from the final price. Let x_{final} be that price, and x_{winner} the price called out by the provider who won the auction. Also, let x_i be the price called by the generic provider P_i in the chain of defaulting providers. Of course, $x_{winner} <= x_i < x_{final}$. We define the price distance of a given price x_i as $d_i = x_{final} - x_i$. According to this definition, d_{winner} is the highest among price distances, and will represent the *overall penalty* to be proportionally shared among defaulting providers. Also, let us define the penalty coefficient as:

$$c_i = \frac{d_i}{\sum_{i=1}^{n} d_i} \tag{1}$$

The reader may notice that the summation of coefficients is equal to 1. Finally, the portion of penalty each defaulting provider will have to pay is calculated as:

$$p_i = c_i \times d_{winner} \tag{2}$$

In the end, the provider who will serve the demand is also awarded the amount deriving from the overall penalty (d_{winner}), but the price for serving the demand (which is due by the customer) will be that called by the bidder who was appointed the auction's winner (x_{winner}).

Let us make an explanatory example. Suppose provider P_1 wins the auction calling a price $x_1 = 2$, but is defaulting. Then, the next best bidder P_2 will be

selected, who called out the price $x_2 = 5$. Again, since P_2 is not able to honor, P_3 who called price $x_3 = 7$ is selected. The overall due penalty is $d_{winner} = x_{final} - x_{winner} = 5$, to be shared among defaulting providers P_1 and P_2 in the following way:

$$p_1 = c_1 \times d_{winner} = \frac{d_1}{d_1 + d_2} \times d_1 = 3.125$$
$$p_2 = c_2 \times d_{winner} = \frac{d_2}{d_1 + d_2} \times d_1 = 1.875$$

while the final price to which the request will be served is $x_1 = 2$. In conclusion, the described mechanism penalizes more the bidders that behaved more aggressively during the auction, preserves the customer by granting them the auction's official winning price and awards the bidder who eventually will serve the customer's demand.

4 An adaptive strategy for cloud providers

In this section we focus on the definition of an adaptive strategy that the provider may use when participating to procurement auctions. By strategy we mean a set of rules producing the decisions a provider must take to maximize its own business objective. According to the literature, the behavior of an auction's participant is mainly driven by the information the participant has on the value of the good being sold [10]. If we better analyze the context of cloud auctions, a computing resource can be seen as a good whose actual value (price) is common to all providers, but the estimate E_{pi} of the i-th provider for a given good may differ from the the estimate E_{pj} of the j-th provider according to the diverse needs each provider may have in pursuing their own business objective. The objective of a strategy is to suggest the provider the price to call for the next bid. In calling a price, a strategy may be more or less "aggressive", i.e., may propose higher or lower discounts. The strategy is adaptive, in the sense that is able to adapt the aggressiveness according to a list of *factors*. Providers may then tune their aggressiveness by adequately weighting factors according to their own business needs. Recalling a formula presented in [11], the adaptive strategy will suggest the next bid as:

$$bid = \frac{n-1}{n-(1-\alpha)} \times lastWinningBid \tag{3}$$

where n is the number of bidders participating in the auction and $lastWinningBid$ is the price offered by the bid that won the last round. In case of single-round auctions, $lastWinningBid$ will be the auction's starting price. The parameter α is calculated as follows:

$$\alpha = w_1 \times \frac{P_a}{P_f} + w_2 \times \frac{T_{vm}}{T_{max}} + w_3 \times \frac{L}{L_{max}} + w_4 \times H(t) \tag{4}$$

Each parameter is weighted by a factor (w_1,w_2,w_3,w_4), whose summation gives 1. The formula in 4 was presented in our previous work [7]. Here we briefly recall the meaning of the parameters. $\frac{P_a}{P_f}$ is the ratio between the resource's starting price in the auction and the corresponding price in the standard fixed-price market. $\frac{L}{L_{max}}$ represents the ratio between the time period for which the computing resource is requested and the maximum time period for which a resource can be requested. $\frac{T_{vm}}{T_{max}}$ is the ratio between the computing power demanded by the request and the computing power of the most powerful resource. Finally the $H(t)$ is the current utilization of the host on which the customer task to serve will be scheduled. Different combinations of weights lead to different strategies. When participating in an auction, providers will be guided by the strategy to:

- check the availability of resources required to serve the demand;
- check if the price called by the auctioneer is higher than than the lower bound price [3];
- calculate the bid;
- send the bid to the auctioneer.

In the case of multi rounds auctions, this mechanism is iteratively repeated. If no offer arrives within a round, the good will be assigned to the last round's best offer.

As stressed earlier in Section 3, this mechanism prevents the providers from committing their overall capacity. To face this issue, providers may decide to **overbook** resources, trying to acquire more requests than they are able to serve. The strategy and the formula to evaluate the bids have been rearranged in order to account for the overbooking. The formula for calculating the α parameters becomes:

$$\alpha = w_1 \times \frac{P_a}{P_f} + w_2 \times \frac{T_{vm}}{T_{max}} + w_3 \times \frac{L}{L_{max}} + w_4 \times O(t) \tag{5}$$

where $O(t)$ is the ratio between the auctions lost by the provider while performing the overbooking, and the total number of the auctions in which they participated. According to this parameter, the provider is more aggressive when the won auctions decrease, while they will be more conservative when the won auctions increase. Further, to estimate the convenience of participating in an auction, the provider performing the overbooking will not have to check if resources are available, but will consider the amount of resources which remained unused in the past, and accordingly compute the number of concurrent auctions in which they may compete. As mentioned earlier, penalties must be carefully monitored. A provider may decide to inhibit the overbooking mechanism when the ratio between penalties and gains exceeds a customized threshold.

[3] The lower bound price is specific to the provider. It indicates the minimum price at which the provider is willing to sell the resource

5 Implementation and testing

To assess the viability of the proposed approach a simulator of the designed market has been implemented. The objective was to define a tool capable of simulating a) the procurement auction processes, b) the behavior of the participating providers and c) the arrival of customers' demands of VMs. Tests conducted on simulator were aimed at monitoring the utilization level of providers' datacenters and the responsiveness of the providers' strategies to the declared business objective.

Architectural details of the simulator. The Cloudsim tool [4] has been used to implement the simulation environment where procurement auctions are run. In addition to the existing Cloudsim components, a new component called *Auctioneer* has been introduced. It cooperates with the Cloudsim *Broker* to manage auctions for cloud applications. The Cloudsim *Datacenter* component has been extended to add functions for a)reserving the resources needed to serve a request, b) estimating bids and b)implementing the overbooking mechanism. Also, the *AdaptiveStrategy* class has been implemented which models the strategy providers may adopt. Finally, the Cloudsim *Cloudlet* component, which represents the task submitted by a customer to Cloudsim, has been extended to include features such as the duration of the requested service, the submission time of the demand, the type of the requested VM, and all the necessary information needed to analyze the data extracted from the simulator for statistical purposes.

Characterization of the customers' demand. To characterize the customers' demand for computing capacity, the same pattern of requests reported in Google's cluster data trace [9] has been reproduced. The trace file stores usage information collected during a 29-day period in the month of May 2011 in one of Googles production cluster cell composed of about 12K machines. In particular, we have reproduced the same workload of Google's trace (in terms of jobs and tasks) and used it to simulate the customer's demand in the procurement market. The reason behind this choice is that the Google cluster's workload is characterized by machine requests which range from a few minutes to one-day usage. We believe such workload characterization may be a good candidate to model the customers' demand for short-term VMs, which providers may be willing to serve with their residual capacity (spare pool of VMs). Actually, the customers' demand to submit to the procurement market was obtained by filtering out all the Google workload's micro requests falling behind the hour usage.

The types of requests appearing in the trace have been mapped onto their equivalent Amazon's virtual machine types. In the following list the characteristics of those machines are reported along with the workload percentage of each VM type with respect to the overall daily workload:

- General purpose
  - m1.small - 32/64-bit architecture, 1 vCPU, 1 CU, 1.7GB RAM, 160GB Storage, Low Bandwidth (workload % = 0.6)

- m1.medium - 32/64-bit architecture, 1 vCPU, 2 CU, 3.75GB RAM, 410GB Storage, Moderate Bandwidth (workload % = 0.3)
- m1.large - 64-bit architecture, 2 vCPU, 4 CU, 7.5GB RAM, 820GB Storage, Moderate Bandwidth (workload % = 56)
- m1.xlarge - 64-bit architecture, 4 vCPU, 8 CU, 15GB RAM, 1.6TB Storage, High Bandwidth (workload % = 7)
- m3.xlarge - 64-bit architecture, 4 vCPU, 13 CU, 15GB RAM, 0 Storage, Moderate Bandwidth (workload % = 0.1)

– Compute optimized
  - c1.medium - 32/64-bit architecture, 2 vCPU, 5 CU, 1.7GB RAM, 350GB Storage, Moderate Bandwidth (workload % = 28.9)

– Memory optimized
  - m2.xlarge - 64-bit architecture, 2 vCPU, 6.5 CU, 17.1GB RAM, 420GB Storage, Moderate Bandwidth (workload % = 7.1)

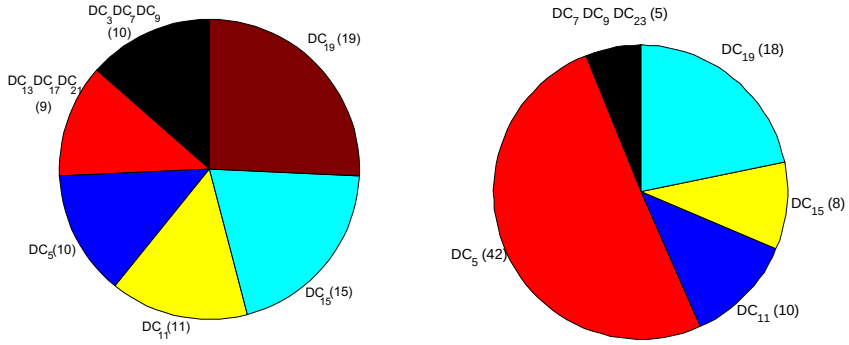
Features of the datacenters. To test the adaptive strategy, we created a set of 24 Datacenters, of which 22 adopt the proposed adaptive strategy and 2 adopt a *Random strategy*. The latters make bids like the formers, with the difference that for them the α parameter is assigned random values in the [0,1] range (they have no specific business objective to pursue). Each Datacenter is provided with 60 physical machines (hosts) equipped with 64 cores, 60 hosts equipped with 128 cores, 60 hosts equipped with 256 cores and 60 hosts equipped with 512 cores, for an overall computing power of 56K cores. Features of Datacenters have been chosen in such a way that all the Datacenters participating in the procurement market will be to sustain the earlier discussed workload.

Provider ID	Strategy	w_1	w_2	w_3	w_4	Overbooking	Provider ID	Strategy	w_1	w_2	w_3	w_4	Overbooking
PR1	Adaptive	0.7	0.1	0.1	0.1	No	PR13	Adaptive	0.4	0.1	0.1	0.4	No
PR2	Adaptive	0.7	0.1	0.1	0.1	Yes	PR14	Adaptive	0.4	0.1	0.1	0.4	Yes
PR3	Adaptive	0.1	0.7	0.1	0.1	No	PR15	Adaptive	0.1	0.4	0.4	0.1	No
PR4	Adaptive	0.1	0.7	0.1	0.1	Yes	PR16	Adaptive	0.1	0.4	0.4	0.1	Yes
PR5	Adaptive	0.1	0.1	0.7	0.1	No	PR17	Adaptive	0.1	0.4	0.1	0.4	No
PR6	Adaptive	0.1	0.1	0.7	0.1	Yes	PR18	Adaptive	0.1	0.4	0.1	0.4	Yes
PR7	Adaptive	0.1	0.1	0.1	0.7	No	PR19	Adaptive	0.1	0.1	0.4	0.4	No
PR8	Adaptive	0.1	0.1	0.1	0.7	Yes	PR20	Adaptive	0.1	0.1	0.4	0.4	Yes
PR9	Adaptive	0.4	0.4	0.1	0.1	No	PR21	Adaptive	0.25	0.25	0.25	0.25	No
PR10	Adaptive	0.4	0.4	0.1	0.1	Yes	PR22	Adaptive	0.25	0.25	0.25	0.25	Yes
PR11	Adaptive	0.4	0.1	0.4	0.1	No	PR23	Random					No
PR12	Adaptive	0.4	0.1	0.4	0.1	Yes	PR24	Random					Yes

Table 1. Weight Setting for the Datacenters' strategies

The 22 Datacenters have been split into two sets, of which only one makes use of overbooking. The weights characterizing the α parameter are shown in Table 1. As the reader may notice, strategies were expressly split in *unbalanced*, for which Datacenters point on just one or two factors, and *balanced*, for which all the weights are assigned the same value. The objective of the simulation is to show that strategies actually guide Datacenters in the choice of the tasks to compete for.

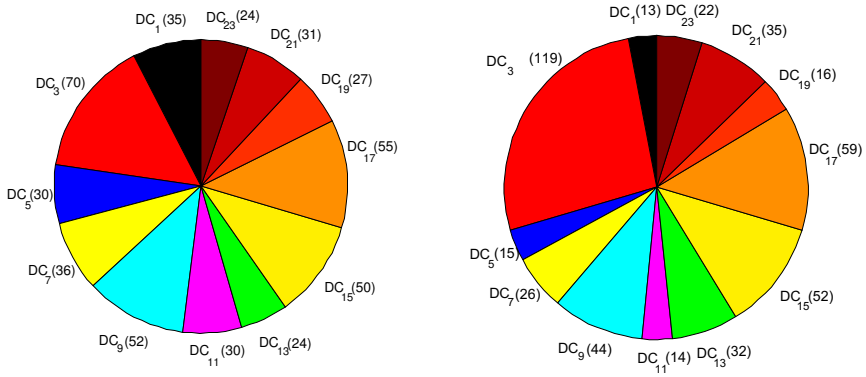
In the tests, the spare resources which providers use to compete in auctions are 20% of their overall resources; the remaining 80% is sold in the traditional fixed-price market. In the context of the simulations we are going to interchangeably use the terms Providers and Datacenters.



(a) ER auction (b) SPSB auction

Fig. 3. Number of 23h-long VMs obtained by Datacenters

Experiments. We ran two different simulations where the workload defined above is submitted to the procurement market. In the first simulation the broker decided to use the ER mechanism to allocate the providers' computing capacity, while in the second the SPSB was used. In the following, results from the two simulations are shown.



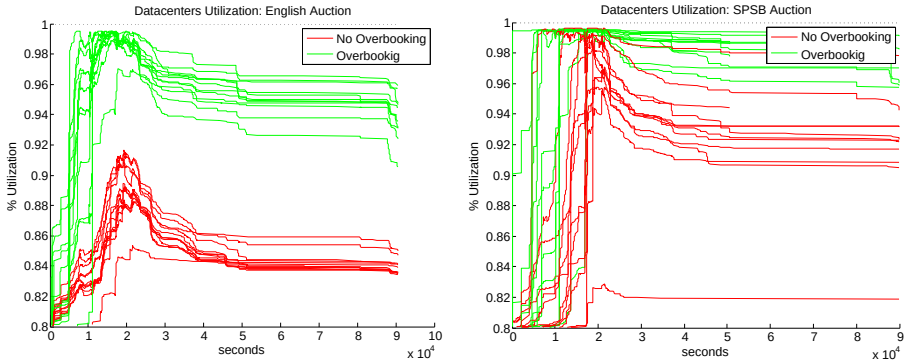
(a) ER auction (b) SPSB auction

Fig. 4. Number of m2.large VMs obtained by Datacenters

The simulations demonstrate that each provider, by properly sizing the weights of their strategy, is able to achieve the chosen objective. In Figures 3(a) and 3(b) we report the number of VM instances having a duration of 23 hours obtained by each Datacenter, respectively in the simulation of the ER and SPSB auction.

Datacenters #5, #11, #15, #19 succeed in pursuing the objective of acquiring a high number of VMs; the reader may notice in Table 1 that those Datacenters have a strategy which points to win auctions where long-lasting VMs are sold. In Figures 4(a) and 4(b) we report the number of VM instances of the VM type *m2.xlarge* (which is the largest among VM types) obtained by Datacenters in the two simulations. It may be noticed in Table 1 that Datacenters #3, #9, #15 and #17 adopt a strategy pointing on large-sized VM, and in fact won a large number of *m2.xlarge* VMs.

One of the most interesting performance indexes is the **host utilization**. All providers aim to achieve the maximum utilization of their data centers. Providers will be willing to call lower bids in order to gain the right to sell the VMs needed to increase the occupancy of their data centers, since the marginal gain from these resources will be certainly high and so it is worth being more aggressive. One may argue that a higher monetary gain may be obtained by selling a few resources at a higher price than selling lots of resources at a very discounted price. But again, the capability of maximizing the gain is out of the scope of this work. Again, what we have proposed is a tool to define, customize and enforce a strategy.



(a) ER auction (b) SPSB auction

Fig. 5. Datacenters utilization

As mentioned above, depending on the roughness of competition, the goal of maximizing the data center's occupancy is roughly pursuable for providers. The overbooking strategy is then necessary to raise the utilization level. Figures 5(a) and 5(b) show the utilization of Datacenters in the ER and the SPSB simulations. The figure depicts the result of a simulation where only the first six hours of workload were simulated, while the utilization is observed for the

24 hours. Those who used overbooking have been depicted in green, while those who did not perform overbooking have been depicted in red. It may be noticed that in the ER simulation the performance of those who used overbooking is much better. This is due by the fact that in multi-round auctions (like the ERs) Datacenters are engaged in long lasting auctions; as a consequence resources are reserved for longer periods, so the overbooking if of much help. In a single-round auction like the SPSB the overbooking mechanism does not bring any evident benefit on the utilization.

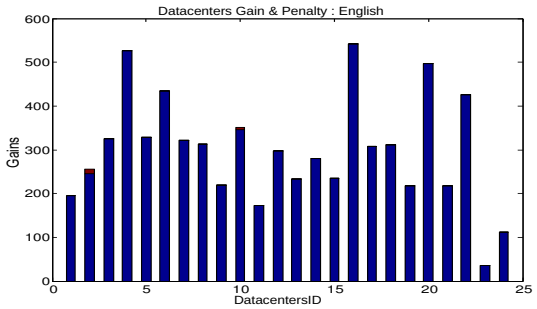


Fig. 6. Gains and penalties of Datacenters in ER auction

Datacenters calling on overbooking must also consider the exposition to penalties in the case they are out of resources when the auction is cleared. With respect to the ER case, the simulations showed that, on average, datacenters making use of overbooking have an advantage also in terms of gains, despite the payment of the penalty, as depicted in Figure 6. In the graph, the red piece of the bar is the amount of incurred penalties, while the blue is the net gain. If we make a two-by-two comparison of Datacenters adopting the same strategy (#1 vs #2, #3 vs #4, and so on) the overbookers on average outperform the non-overbookers.

6 Conclusion

Commercial cloud providers are making huge profits from leasing their computing capacity to requesting customers. Cloud resources are mainly allocated through the direct-sell pricing model which has been proved to be economically inefficient. Further, this pricing strategy prevents providers from allocating their full computing capacity, thus causing a residual capacity to remain unsold. Alternative pricing schemes should then be sought that might help Cloud providers to increment their profit. In this paper, we proposed the design of an open market of cloud resources, where the residual computing capacity of providers is allocated through procurement auctions. An adaptive strategy was also devised that, suitably tailored to the provider's business objective, helps them to maximize the revenue in the context of procurement auctions. Tests conducted on a simulator showed the viability of the proposal.

References

1. Agmon Ben-Yehuda, O., Ben-Yehuda, M., Schuster, A., Tsafrir, D.: Deconstructing amazon ec2 spot instance pricing. In: 2011 IEEE Third International Conference on Cloud Computing Technology and Science (CloudCom). pp. 304–311 (2011)
2. Bonacquisto, P., Di Modica, G., Petralia, G., Tomarchio, O.: Procurement auctions to maximize players utility in cloud markets. In: CLOSER 2014 - Proceedings of the 4th International Conference on Cloud Computing and Services Science. Barcelona (Spain) (Apr 2014)
3. Buyya, R., Yeo, C.S., Venugopal, S.: Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. In: 10th IEEE International Conference on High Performance Computing and Communications (HPCC'08). pp. 5 –13 (Sep 2008)
4. Calheiros, R., Ranjan, R., Beloglazov, A., De Rose, C.A., Buyya, R.: Cloudsim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. In: Software: Practice and Experience (2011)
5. Chard, K., Bubendorfer, K.: High Performance Resource Allocation Strategies for Computational Economies. IEEE Trans. Parallel Distrib. Syst. 24(1), 72–84 (Jan 2013)
6. Cramton, P., Shoham, Y., Steinberg, R.: Combinatorial auctions. The MIT Press (2005)
7. Di Modica, G., Petralia, G., Tomarchio, O.: Procurement auctions to trade computing capacity in the Cloud. In: 8th Int. Conf. on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC 2013). Compiegne (France) (Oct 2013)
8. Di Modica, G., Tomarchio, O.: Matching the business perspectives of providers and customers in future cloud markets. Cluster Computing pp. 1–19 (2014)
9. Google: Traces of google workloads. http://code.google.com/p/googleclusterdata/ (2011)
10. Klemperer, P.: Auction Theory: A Guide to the Literature. Journal Of Economic Surveys 13(3) (1999)
11. McAfee, R. P. e McMillan, J.: Auctions and bidding. Journal of Economic Literature 15, 699–738 (1987)
12. Parsons, S., Rodriguez-Aguilar, J.A., Klein, M.: Auctions and bidding: A guide for computer scientists. ACM Computing Surveys 43(2) (Feb 2011)
13. Risch, M., Altmann, J., Guo, L., Fleming, A., Courcoubetis, C.: The GridEcon Platform: A Business Scenario Testbed for Commercial Cloud Services. In: Grid Economics and Business Models, vol. 5745, pp. 46–59. Springer (2009)
14. Samimi, P., Teimouri, Y., Mukhtar, M.: A combinatorial double auction resource allocation model in cloud computing. Information Sciences (2014), in Press
15. Smeltzer, L.R., Carr, A.: Reverse auctions in industrial marketing and buying. Business Horizons 45(2), 47–52 (2002)
16. Sulistio, A., Kim, K.H., Buyya, R.: Managing Cancellations and No-Shows of Reservations with Overbooking to Increase Resource Revenue. In: 8th IEEE International Symposium on Cluster Computing and the Grid (CCGRID'08). pp. 267–276 (May 2008)
17. Vinu Prasad, G., Rao, S., Prasad, A.: A Combinatorial Auction mechanism for multiple resource procurement in cloud computing. In: Intelligent Systems Design and Applications (ISDA), 2012 12th International Conference on (2012)
18. Wang, Q., Ren, K., Meng, X.: When cloud meets ebay: Towards effective pricing for cloud computing. In: INFOCOM, 2012 Proceedings IEEE. pp. 936–944 (2012)