

Combinatorial Auction-Based Mechanisms for VM Provisioning and Allocation in Clouds

Sharrukh Zaman
Department of Computer Science
Wayne State University
Detroit, MI 48202
Email: sharrukh@wayne.edu

Daniel Grosu
Department of Computer Science
Wayne State University
Detroit, MI 48202
Email: dgrosu@wayne.edu

Abstract—Current cloud providers use fixed-price based mechanisms to allocate Virtual Machine (VM) instances to their users. The fixed-price based mechanisms do not provide an efficient allocation of resources and do not maximize the revenue of the cloud providers. A better alternative would be to use combinatorial auction-based resource allocation mechanisms. In this PhD dissertation we will design, study and implement combinatorial auction-based mechanisms for efficient provisioning and allocation of VM instances in cloud computing environments. We present our preliminary results consisting of three combinatorial auction-based mechanisms for VM provisioning and allocation. We also present an efficient bidding algorithm that can be used by the cloud users to decide on how to bid for their requested bundles of VM instances.

Index Terms—cloud computing; virtual machine allocation; mechanism design; combinatorial auctions;

I. INTRODUCTION

Cloud computing systems provide the next computing infrastructure enabling users to provision remote resources for their computational needs, eliminating the upfront costs of setting up their own systems. Clouds give users the illusion of an infinite computing resource available on demand and allow them to acquire and pay for resources on a short term basis. Examples of cloud computing systems include both commercial (e.g., Microsoft Azure [1], Amazon EC2 [2]) and open source ones (e.g., Eucalyptus [3]). The resource allocation and trading mechanisms used by the current cloud computing systems are inefficient and inflexible due to flat rate pricing. Recently, Amazon, one of the major cloud computing providers, recognized these issues and introduced a simple auction mechanism, called Spot Instances, for selling unused EC2 resources [4].

We argue that the fixed price-based resource allocation and trading mechanisms currently in use in cloud computing systems do not provide an efficient allocation of resources and do not maximize the revenue of the cloud providers. A better alternative would be to use auction-based resource allocation mechanisms. This argument is supported by the economic theory; when the auction costs are low, as is the case in the context of cloud computing, auctions are especially efficient over the fixed-price markets since products are matched to customers having the highest valuation [5]. We go a little bit further and argue that the most suitable mechanisms for

resource allocation and pricing in clouds should be based on *combinatorial auctions*. In combinatorial auctions [6] the participants bid on combinations of discrete items rather than just individual items. These auctions have clear advantages over other auction types when the auctioned items are complementary, that is when the auctioned items have a higher value as a set than as separate parts. An example of complementary products are CPUs and memory. Obtaining only CPUs without memory has no value to the user, but obtaining both CPUs and memory has value. Another example is when a cloud provider offers several types of Virtual Machine (VM) instances (large, intermediate and small) and a user requires a combination of one large instance and three small instances in order to run her application and meet the quality of service (QoS) requirements. Obtaining only the large instance or the three small instances has no value for the user, but obtaining the bundle composed of one large and three small instances has the highest value for the user, allowing her to meet the QoS requirements. A major advantage of combinatorial auctions over simple auctions like first-price and second-price auctions is that bidders can express more fully their preferences. Allowing the bidders to express more complex valuations leads to improved efficiency (i.e., the items are allocated to the users who value them most) and greater revenues. In recent years, combinatorial auctions have been successfully applied to several problems such as selling wireless spectrum, transportation, and industrial procurement [6]. This confirms their power in expressing complex preferences and providing efficient resource allocation.

The *objective of this Ph.D. dissertation* is to design, study and implement combinatorial auction-based mechanisms for efficient provisioning and allocation of VM instances in cloud computing environments. The central hypothesis is that allocation efficiency and revenue maximization can be obtained by inducing users to fully express and truthfully report their preferences to the system. The *rationale* for our research is that, once efficient resource provisioning and allocation mechanisms that take into account the incentives of the users and cloud providers are developed and implemented, it will become more efficient to utilize cloud computing environments for solving challenging problems in business, science and engineering.

The problem of VM provisioning and allocation in clouds may take different forms. The first and simplest form assumes that the VM instances are pre-provisioned and that the number of different types of available VM instances is fixed. In this setting users bid for their required bundles and a combinatorial auction-based mechanism determines only the allocation and payment and not the way the resources are provisioned. The second form assumes dynamic provisioning. In this setting the combinatorial auction takes place first and based on the auction outcome the mechanism decides on the provisioning and allocation of VMs and also on the user payments. Another dimension that brings complexity into this problem is the consideration of time. A simpler approach to incorporate time is to run auctions at fixed intervals and allocate resources for ‘one unit of time’. In this case, the users participate in consecutive auctions until their task is completed. A more complex model enables users to include time with their bids. In this case, the system guarantees that if a user wins, she will be allocated the resources for the entire period she requested. Our aim is to solve all the above problems and have a complete research on VM provisioning and allocation in clouds using combinatorial auction-based mechanisms. This will enable a cloud provider to choose one or more mechanisms from the set of the proposed efficient mechanisms based on their system requirement and service goals.

II. SIGNIFICANCE OF OUR RESEARCH

The existing cloud resource allocation and trading is based on a fixed-price model which does not lead to efficient resource allocation and does not provide incentives to the users to decrease the use of resources during heavily loaded periods. The fundamental issue is the lack of mechanisms that take into account the users incentives and allow them to fully express their preferences. Mechanism design [7], a subfield of game theory, addresses the problem of how to design a system (via price incentives and payments) to induce desired agent behavior and achieve system-wide goals. Mechanism design and auction theory have been applied in a variety of different contexts such as web caching, task allocation, and internet routing [8]. As a compelling example, over the period from 1993 to 1997, the FCC used mechanism design techniques to design spectrum auctions which yielded \$23 billion in winning bids [9]. Previous spectrum auctions are estimated to have led to speculation and loss of valuation in hundreds of millions of dollars. Thus, mechanism design and auction theory have had an enormous impact on the allocation of traditional resources and we anticipate that they will have the same impact on the allocation of computational resources as well.

Our dissertation research is significant because it is expected to facilitate efficient execution of applications in cloud computing environments by providing resource allocation and trading mechanisms that maximize the revenue of the cloud providers and reduce the waste of energy and computational resources. The proposed research is multidisciplinary, drawing on theories from computer and information sciences, game

theory and economics. We expect that the newly developed combinatorial auction-based VM provisioning and allocation mechanisms and their software implementation will deliver significant performance and efficiency of computation in current and future cloud computing systems. These mechanisms are not only intended for commercial clouds, but also for private clouds. An organization that wants to utilize its private cloud efficiently would also require efficient provisioning and allocation mechanisms. By introducing virtual money and auction-based mechanisms, it is possible to ensure that different units/departments of a large business entity use the central computing resources according to their requirements and the perceived value of their work. It is worth mentioning here that first reported work on auction-based allocation of computing resources was developed for an academic environment, where an auction-based mechanism was used to allocate processor time on a single computer [10]. Considering all of the above, we envision combinatorial auction-based provisioning and allocation mechanisms as an integral part of future cloud computing systems. The potential impact of our research is a reduction in the waste of computational resources and energy in these organizations.

Our research fits within the area of economics of Cloud computing which is highly relevant to the CCGrid audience. In fact, past CCGrid conferences had at least one session dedicated to the economics of clouds and grids and several researchers in this area are regular CCGrid participants.

III. RELATED WORK

The use of auctions in computing dates back to 1968 when Sutherland [10] proposed allocating the processor time in a single computer via auctions. Gagliano [11] also investigated allocating computing resources through auctions, where the tasks themselves are provided enough intelligence to calculate the bid that is necessary to get the required resources. Recently, researchers investigated economic models for resource allocation in computational grids. Wolski *et al.* [12] compared commodities markets and auctions in grids in terms of price stability and market equilibrium. Gomoluch and Schroeder [13] simulated the double auction protocol for resource allocation in grids and showed that it outperforms the conventional round-robin approach. Das and Grosu [14] proposed a combinatorial auction-based protocol for resource allocation in grids. They considered a model where different grid providers can provide different types of computing resources. An ‘external auctioneer’ collects these information about the resources and runs a combinatorial auction-based allocation mechanism where users participate by requesting bundles of resources. The major difference between the present work and the one presented in [14] is that we are considering allocation of VM instances of a single cloud provider whereas in [14], the authors considered the problem of allocating different types of physical resources from multiple grid providers.

Currently researchers are investigating the economic aspects of cloud computing from different points of view. Wang *et al.* [15] studied different economic and system implications

of pricing resources in clouds. Altmann *et al.* [16] proposed a marketplace for resources where the allocation and pricing are determined using an exchange market of computing resources. In this exchange, the service providers and the users both express their ask and bid prices and matching pairs are granted the allocation and removed from the system. In [17], a testbed for cloud services was designed to be able to test different mechanisms on clouds. The authors deployed the exchange mechanism described in [16] on this platform. We are considering combinatorial auction mechanisms instead of an exchange in our research. In this case, the cloud providers need not express any valuations of the items (or ask prices). Also, combinatorial auctions are more useful in the cloud computing context since usually users require a particular bundle of resources in order to run their applications.

The tool CloudCmp [18] was developed to assist users choose the appropriate service providers based on the user's requirements. Buyya *et al.* [19] proposed an infrastructure of federated clouds for auction-based resource allocation across multiple clouds. Researchers also investigated whether cloud solutions are economically feasible for all kinds of computing needs. Walker *et al.* [20] proposed a model to determine the benefits of acquiring storage services from clouds.

A detailed survey on combinatorial auctions can be found in [21]. The book by Cramton *et al.* [6] provides good foundational knowledge on this topic. Lehmann *et al.* [22] studied combinatorial auctions with single-minded bidders and devised a greedy mechanism for combinatorial auctions. In our thesis, we extend this mechanism in order to solve the VM allocation and pricing in clouds problem. Archer *et al.* [23] considered another case of single-minded bidders where multiple identical copies are available for different types of items. They provided a mixed integer programming based algorithm for winner determination and showed theoretically that their solution performs better than generalized solutions for this special case. We extend this mechanism such that it can be used to solve the VM allocation and pricing problem we consider in our thesis.

IV. RESEARCH ACCOMPLISHMENTS

In this section, we present our research accomplishments over the last two years.

In our paper presented at IEEE CloudCom 2010 [24], we formulated the Virtual Machine Allocation Problem (VMAP) and designed two combinatorial auction-based mechanisms to solve it. This paper received the IEEE CloudCom 2010 Best Student Paper Award. VMAP assumes pre-provisioned VM instances. In this model the cloud provider offers m different types of VM instances $VM_1, \dots, VM_m$ whose computing power is characterized by 'weights' $w_1, \dots, w_m$. There are k_i instances of type VM_i available for allocation and n users who request bundles of VM instances through 'bids' submitted to a combinatorial auction-based mechanism. The bid submitted by user u_j is denoted by $B_j = (r_1^j, \dots, r_m^j, v_j)$, where r_i^j is the number of VM_i instances requested and v_j is the maximum amount user u_j wants to pay for the bundle.

The bids are submitted to a combinatorial auction-based mechanism that runs periodically. The mechanism computes a set of winners W from the users and a payment p_j for each user. The goal of an efficient mechanism is to maximize the sum of the valuations of the winning bidders, i.e., $\max \sum_{j: u_j \in W} v_j$ while satisfying the resource availability constraints. The utility of a winning bidder is $v_j - p_j$, i.e., the value she gets by winning her bundle minus the amount she pays for it. The utility of a losing bidder is zero. The allocation mechanism must be *truthful*, that is, the utility of a user is maximized only when she reports her true valuation v_j to the mechanism. This is an important feature that will prevent users from manipulating the mechanism since users will not have incentives to submit bids that do not represent their true required bundle and its true valuation.

VMAP is an NP-hard problem, therefore we devised two combinatorial auction-based approximation mechanisms to solve it. We called the first mechanism CA-GREEDY, since it is based on a greedy approach to solve VMAP. CA-GREEDY extends the mechanism proposed by Lehmann *et al.* [22] so that it can solve VMAP. CA-GREEDY orders the user bids according to their 'bid density', which is the ratio $v_j / \sqrt{s_j}$, where v_j is user u_j 's reported valuation and s_j is the total weight of the bundle requested by u_j (i.e., $s_j = \sum_{i=1}^m w_i r_i^j$). CA-GREEDY then allocates VM instances according to this order, as long as enough resources are available. The mechanism determines the payment p_j that a winner u_j needs to pay by multiplying $\sqrt{s_j}$ with the highest bid density among the losing players who would win if u_j would not be a winner. We proved that CA-GREEDY is a $\sqrt{M}$ -approximation solution to VMAP, where M is the total computing power of all the VMs, expressed as the sum of their weights. We also proved that it is a truthful mechanism.

The second mechanism we designed to solve VMAP is called CA-LP, which is based on a mixed integer programming-based approach. We designed this mechanism by extending the mechanism proposed by Archer *et al.* [23] in order to solve VMAP. This mechanism computes the allocation by solving the linear program: $\max \sum_{j=1}^n x_j v_j$, subject to (i) $\sum_{j=1}^n x_j r_i^j \leq k_i, i = 1, \dots, m$; and (ii) $0 \leq x_j \leq 1, j = 1, \dots, n$. Here, $x_j = 1$ means that user u_j wins her bundle, while $x_j = 0$ means that user u_j is a loser. Then, the mechanism employs randomized rounding where user u_j is selected as a winner with probability x_j , if this allocation does not violate any constraint. This operation is executed in order of decreasing x_j so that if there is a violation in the constraints, the user assigned a lower x_j is not included in the set of winners W . The payment is determined by computing the minimum valuation of each winners at which they would win their bid given that the corresponding y_j 's do not change. We proved that CA-LP is truthful in expectation.

We performed extensive simulation experiments to compare CA-LP and CA-GREEDY with a fixed price-based mechanism called FIXED-PRICE, which allocates VM instances on a first-come first-served basis and charges a predetermined price for

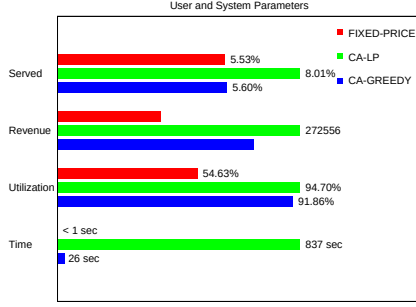


Fig. 1. Overall performance of the mechanisms

them. Our results showed that CA-LP performs the best among the three mechanisms. It allows the completion of 40% more jobs than the FIXED-PRICE mechanism for the same set of users and jobs. CA-LP also generates more than twice the revenue generated by FIXED-PRICE and utilizes better the cloud resources. On the other hand, CA-GREEDY slightly exceeds the percentage of completed jobs, generates about 90% more revenue, and achieve a 68% increase in utilization compared to FIXED-PRICE. CA-GREEDY has a lower execution time than that of CA-LP, and thus, we recommend it for general purpose VM allocation in clouds.

In Figure 1, we show the performance of the three mechanisms. To show different metrics in one plot, the length of the horizontal bars are normalized to the highest value in each category. We see that FIXED-PRICE and CA-GREEDY serve (i.e., complete the task of) more than 5.5% users, while, CA-LP serves 8% of the users. Both CA-GREEDY and CA-LP perform significantly better in terms of revenue and resource utilization. There are several reasons that make the CA-based mechanisms perform better. First, they allocate and price resources based on the market demand, whereas FIXED-PRICE allocates resources only when the bid is greater than the set price. FIXED-PRICE under-utilizes the resources when demand is low and it cannot obtain higher revenue when there is a high demand of resources. Another reason behind the results are the selection of winners: FIXED-PRICE allocates on a first-come first-served basis while CA-mechanisms allocate based on the perceived value of the bids. Therefore, the CA-mechanisms are able to select to serve users such that the revenue obtained by the cloud provider is increased. We can see a reflection of this fact in Figure 2, where we show the revenue generated by the mechanisms for different ranges of user valuations. The horizontal axis shows the lower and upper limit of valuations and the vertical axis shows the revenue generated by each mechanism. We observe two trends here. First, for all ranges, CA-mechanisms generated about twice the revenue generated by FIXED-PRICE. And while the CA-mechanisms' revenue grows almost proportionally to the upper limits of valuations, FIXED-PRICE cannot maintain this.

The above-mentioned mechanisms consider the VM instances to be pre-provisioned, i.e., the number of different types of VM instances are predetermined. Also, they did not consider the costs of running the VMs. In a paper presented at

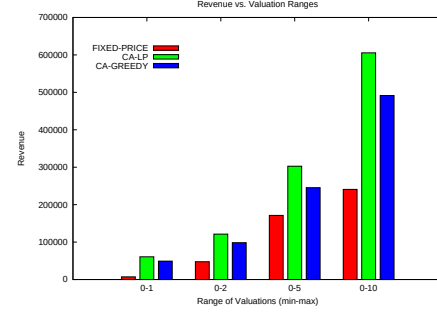


Fig. 2. Effect of valuation ranges on revenue

IEEE CloudCom 2011 [25] and a poster presented at USENIX HotCloud'11 [26] we considered a dynamic provisioning setting. We defined the Dynamic Virtual Machine Provisioning and Allocation Problem (DVMPA) and designed a combinatorial auction-based provisioning and allocation mechanism to solve it. DVMPA assumes that the VM instances are provisioned dynamically based on the combinatorial auction outcome, so that more resources can be allocated efficiently.

We assume that the total weight of the available computing resources is M , which is equal to M instances of VMs of unit weight. We also assume that the costs incurred for an idle and running VM instance of unit weight per an auction interval are c_I and c_R , respectively. As before, there are m VM types $VM_1, \dots, VM_m$ and n users submitting their bids for bundles of VM instances. The mechanism needs to compute the allocation and payment and also determine the vector $k_1, \dots, k_m$, which is the number of each type of VM instances that needs to be provisioned.

Adding the cost parameters leads to determining the profit of the provider as follows.

$$P = \sum_{j=1}^n x_j p_j - c_R \sum_{j=1}^n x_j s_j - c_I \left(M - \sum_{j=1}^n x_j s_j \right)$$

Here, the first term is the total revenue collected from the bidders. The second term represents the cost incurred from the VM instances allocated to users and the third term is the cost due to the idle computing resources. The objective of the mechanism is to maximize the profit subject to the resource availability constraints.

DVMPA is also an NP-hard problem, for which we again considered the design of an approximation mechanism. The mechanism is an extended version of CA-GREEDY, which we call CA-PROVISION. CA-PROVISION computes bid density of each user u_j as v_j/s_j , where s_j is the total weight of the requested bundle. It then allocates bundles according to the sorted order of bid density until all the resources are exhausted. The payment of a winning bidder is calculated as in CA-GREEDY. Finally, the mechanism provisions the resources into the number of VM instances that are determined by the allocation algorithm.

We proved that CA-PROVISION is a truthful mechanism. Since we already showed in [24] that the combinatorial

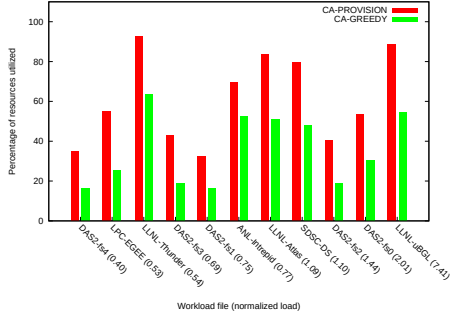


Fig. 3. Resource utilization vs. normalized load

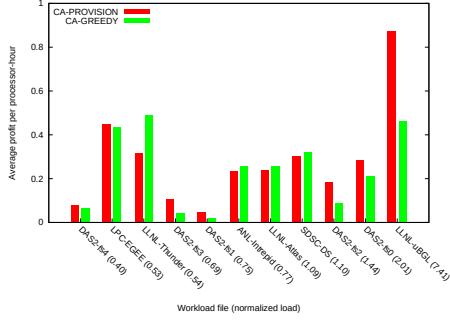


Fig. 4. Average profit per processor-hour vs. normalized load

auction-based mechanisms perform better than the fixed-price mechanisms, we only compared CA-PROVISION with CA-GREEDY in this work. We used traces from the Parallel Workloads Archive [27] to drive the simulations. Our results indicate that CA-PROVISION increases the revenue of the cloud provider for all workload traces. In terms of profit, CA-PROVISION generates higher profit for workload traces that have higher volume of jobs per unit time. For workloads with low job rates, CA-PROVISION generates higher profit for half the workloads and CA-GREEDY generates higher profit for the others. Most importantly, CA-PROVISION significantly improves the resource utilization and the number of completed jobs.

In Figure 3, we show the resource utilization for different workloads by both CA-GREEDY and CA-PROVISION. The horizontal axis shows the workloads ordered by their normalized load, which measures the relative load with respect to available resources. We see that CA-PROVISION always utilizes significantly more resources than CA-GREEDY for all workloads. Especially, CA-PROVISION utilizes almost as twice as many resources as CA-GREEDY for both cases with very high and very low demands. This is due to the dynamic provisioning approach, which provisions resources based on the pattern of the demands. On the other hand, we see in Figure 4 that CA-PROVISION generates higher profit for high demand scenarios. For the other workloads, CA-GREEDY often achieves higher profit when the demand matches the supply.

Bidding in a combinatorial auction is complex and may

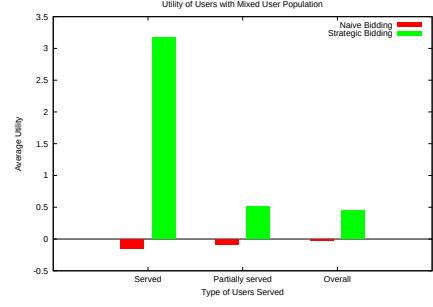


Fig. 5. Average utility of strategic and naive users

become a challenge for successful implementation of combinatorial auction-based mechanisms in clouds. In a paper presented at IEEE CLOUD 2011 [28] we introduced an efficient bidding strategy (EBS) for allocating VM instances for malleable parallel jobs in clouds. The goal of EBS is to generate bids for combinatorial auctions so that the utility of the user is maximized. Since the mechanisms we devised are truthful, the task of EBS is to generate a truthful bid considering the job, the system parameters, and the users' preferences.

Malleable applications are parallel jobs that can run on any number of processors [29]. However, the speedup of a malleable job is not linear with the number of processors due to parallelization overhead. Havill and Mao [29] modeled the execution time of a malleable job using the 'setup time' as the only overhead of parallelization. We extended that model incorporating the overhead associated with the distribution of input data and communication overhead, which may vary between applications. From the parallel execution time we determine the speedup, which is the basis for computing the valuation in EBS.

EBS works as follows. It first finds the optimal number of processors needed to execute the job using the model presented in [29]. It then calculates the best possible bundle of VMs containing this optimal number of processors. Next, our new parallel malleable job speedup model is used to calculate the actual speedup. If the valuation associated with the computed speedup exceeds the budget, the bundle of VM instances is adjusted as follows. Since running the job on VM instances with more processors can be expensive, we replace a 'bigger' VM with an equivalent number of 'smaller' VMs until the resulting valuation falls within the budget range. If this process fails to produce a feasible bid, we decrease the number of processors and start from creating the best bundle for the updated number of processors. Finally, either the best bundle within the budget is generated, or the user does not participate in the mechanism at all.

We performed simulation experiments to compare EBS with a naive bidding strategy. Results show that the average utility of users bidding using EBS far exceeds that achieved by users applying the naive strategy. In Figure 5 we show the average utility received by the strategic and naive bidders when they participate in the same auctions. The results show that in

the presence of both types of users (naive and EBS), users employing EBS for their bids gain higher utility than the naive users. On the other hand, the naive users are getting negative utility on average, because they fail to generate truthful and efficient bids for their intended tasks.

The advantage of EBS is that it takes into account the speedup when calculating the valuation, which is a natural thing to do for a user executing parallel jobs. EBS finds the best bundle within the given budget and reports the true valuation to the mechanism. Since the combinatorial auction-based mechanisms we used are truthful, EBS guarantees that the users will maximize their utilities by employing it in determining their bids.

V. FUTURE WORK

In order to complete our thesis, we will conduct research on more dynamic models than the ones we considered so far. We will consider incorporating time into the user bids so that a user can also specify the time for which she needs the VM bundle. We plan to design truthful mechanisms to solve the problem in this new setting. Another direction of our future research is towards dynamic provisioning of VMs where the users request the configurations based on their specific needs. That is, users can specify their needed types of VMs instead of choosing among the types of VM instances offered by the cloud provider. This will be a very hard problem to solve, but it will provide a more general and more dynamic way to allocate and provision the VM instances. Finally, we would like to address the bidding strategies of the users for each different types of combinatorial auction-based mechanisms we design.

There are several challenges that we need to overcome in order to complete the above mentioned research. First of all, we know that all the models that we will consider will lead to NP-hard problems. Therefore, we need to find approximate solutions for determining the allocation and provisioning for each of the new settings. Finally, we must ensure that the mechanisms we design are truthful, which is difficult to achieve when employing approximate solutions.

ACKNOWLEDGMENT

This work was supported in part by NSF grants DGE-0654014 and CNS-1116787.

REFERENCES

- [1] Microsoft, "Windows Azure platform, <http://www.microsoft.com/windowsazure/>."
- [2] Amazon, "Amazon Elastic Compute Cloud (Amazon EC2), <http://aws.amazon.com/ec2/>."
- [3] D. Nurmi, R. Wolski, C. Grzegorzcyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov, "The eucalyptus open-source cloud-computing system," in *Proc. of the 9th IEEE/ACM Intl. Symp. on Cluster Comp. and the Grid*, May 2009, pp. 124–131.
- [4] Amazon, "Amazon EC2 spot instances," <http://aws.amazon.com/ec2/spot-instances/>.
- [5] R. Wang, "Auctions versus posted-price selling," *The American Economic Review*, vol. 83, no. 4, pp. 838–851, 1993.
- [6] P. Cramton, Y. Shoham, and R. Steinberg, *Combinatorial Auctions*. The MIT Press, 2005.

- [7] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic Game Theory*. Cambridge University Press, 2007.
- [8] J. Feigenbaum and S. Shenker, "Distributed algorithmic mechanism design: Recent results and future directions," in *Proc. of the 6th ACM Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, September 2002, pp. 1–13.
- [9] "The FCC report to congress on spectrum auctions," Federal Communications Commission: Wireless Telecommunications Bureau, Tech. Rep., 1997.
- [10] I. E. Sutherland, "A futures market in computer time," *Communications of the ACM*, vol. 11, no. 6, pp. 449–451, 1968.
- [11] R. A. Gagliano, M. D. Fraser, and M. E. Schaefer, "Auction allocation of computing resources," *Communications of the ACM*, vol. 38, no. 6, pp. 88–102, 1995.
- [12] R. Wolski, J. S. Plank, J. Brevik, and T. Bryan, "Analyzing market-based resource allocation strategies for the computational grid," *Intl. J. of High Performance Comp Appl.*, vol. 15, no. 3, pp. 258–281, 2001.
- [13] J. Gomoluch and M. Schroeder, "Market-based resource allocation for grid computing: A model and simulation," in *Proc. 1st International Workshop on Middleware for Grid Computing*, 2003, pp. 211–218.
- [14] A. Das and D. Grosu, "Combinatorial auction-based protocols for resource allocation in grids," in *Proc. 19th International Parallel and Distributed Processing Symposium, 6th Workshop on Parallel and Distributed Scientific and Engineering Computing*, 2005.
- [15] H. Wang, Q. Jing, R. Chen, B. He, Z. Qian, and L. Zhou, "Distributed systems meet economics: Pricing in the cloud," in *Proc. 2nd USENIX Workshop on Hot Topics in Cloud Computing*, 2010.
- [16] J. Altmann, C. Courcoubetis, G. D. Stamoulis, M. Dramitinos, T. Rayna, M. Risch, and C. Bannink, "GridEcon: A market place for computing resources," in *Proc. Workshop on Grid Economics and Business Models*, 2008, pp. 185–196.
- [17] M. Risch, J. Altmann, L. Guo, A. Fleming, and C. Courcoubetis, "The GridEcon platform: A business scenario testbed for commercial cloud services," in *Proc. Workshop on Grid Economics and Business Models*, 2009, pp. 46–59.
- [18] A. Li, X. Yang, S. Kandula, and M. Zhang, "CloudCmp: Shopping for a cloud made easy," in *Proc. 2nd USENIX Workshop on Hot Topics in Cloud Computing*, 2010.
- [19] R. Buyya, R. Ranjan, and R. N. Calheiros, "InterCloud: Utility-oriented federation of cloud computing environments for scaling of application services," in *Proc. 10th International Conference on Algorithms and Architectures for Parallel Processing*, 2010, pp. 13–31.
- [20] E. Walker, W. Briskin, and J. Romney, "To lease or not to lease from storage clouds," *IEEE Computer*, vol. 43, no. 4, pp. 44–50, 2010.
- [21] S. de Vries and R. V. Vohra, "Combinatorial auctions: A survey," *INFORMS Journal on Computing*, vol. 15, no. 3, pp. 284–309, 2003.
- [22] D. Lehmann, L. I. O'Callaghan, and Y. Shoham, "Truth revelation in approximately efficient combinatorial auctions," *Journal of the ACM*, vol. 49, no. 5, pp. 577–602, 2002.
- [23] A. Archer, C. Papadimitriou, K. Talwar, and E. Tardos, "An approximate truthful mechanism for combinatorial auctions with single parameter agents," *Internet Mathematics*, vol. 1, no. 2, pp. 129–150, 2005.
- [24] S. Zaman and D. Grosu, "Combinatorial auction-based allocation of virtual machine instances in clouds," in *Proc. 2nd IEEE Intl. Conf. on Cloud Computing Technology and Science*, 2010, pp. 127–134.
- [25] —, "Combinatorial auction-based dynamic VM provisioning and allocation in clouds," in *Proc. 3rd IEEE Intl. Conf. on Cloud Computing Technology and Science*, 2011, pp. 107–114.
- [26] —, "Dynamic VM provisioning and allocation in clouds: A combinatorial auction-based approach," Poster Presented at 3rd USENIX Workshop on Hot Topics in Cloud Computing (HotCloud'11), 2011.
- [27] D. G. Feitelson, "Parallel Workloads Archive: Logs," <http://www.cs.huji.ac.il/labs/parallel/workload/logs.html>.
- [28] S. Zaman and D. Grosu, "Efficient bidding for virtual machine instances in clouds," in *Proc. 4th IEEE International Conference on Cloud Computing (CLOUD 2011)*, 2011.
- [29] J. T. Havill and W. Mao, "Competitive online scheduling of perfectly malleable jobs with setup times," *European J. of Oper. Res.*, vol. 187, no. 3, pp. 1126–1142, 2008.