# The Effects of Untruthful Bids on User Utilities and Stability in Computing Markets

Sergei Shudler, Lior Amar, Amnon Barak
*Department of Computer Science*
*The Hebrew University of Jerusalem*
*Jerusalem, 91904 Israel*
*Email: sergshu@gmail.com*

Ahuva Mu'alem
*Social and Information Sciences Laboratory*
*California Institute of Technology*
*Pasadena, California, USA 91125*

***Abstract*—Markets of computing resources typically consist of a cluster (or a multi-cluster) and jobs that arrive over time and request computing resources in exchange for payment. In this paper we study a real system that is capable of preemptive process migration (i.e. moving jobs across nodes) and that uses a market-based resource allocation mechanism for job allocation. Specifically, we formalize our system into a market model and employ simulation-based analysis (performed on real data) to study the effects of users' behavior on performance and utility. Typically online settings are characterized by a large amount of uncertainty; therefore it is reasonable to assume that users will consider simple strategies to game the system. We thus suggest a novel approach to modeling users' behavior called the Small Risk-aggressive Group model. We show that under this model untruthful users experience degraded performance. The main result and the contribution of this paper is that using the $k$-th price payment scheme, which is a natural adaptation of the classical second-price scheme, discourages these users from attempting to game the market. The preemptive capability makes it possible not only to use the $k$-th price scheme, but also makes our scheduling algorithm superior to other non-preemptive algorithms. Finally, we design a simple one-shot game to model the interaction between the provider and the consumers. We then show (using the same simulation-based analysis) that market stability in the form of (symmetric) Nash-equilibrium is likely to be achieved in several cases.**

***Keywords*-Strategic behavior; Resource allocation; Market-based scheduling; Market stability; Pricing**

## I. Introduction

In recent years the area of High-Performance Computing (HPC) has witnessed an increasing amount of studies of market-based resource allocation mechanisms. These mechanisms incorporate market features into their design and allocate scarce resources more efficiently, not only by increasing utilization (in the traditional sense) but also by maximizing users' utilities [1]–[6]. One of the central principles in these mechanisms is *incentive compatibility*; i.e., the ability to discourage strategic users from gaming the market to obtain better personal utility for themselves [7]. By gaming the market these users disproportionally harm other users who are best described as "conservative" in terms of gaming attempts. Usually the goal of incentive compatible mechanisms is to maximize the utility for all the users such that gaming the market will not increase the utility of any of the users. However, there are costs associated with using incentive compatible mechanisms. For example, in the combinatorial auctions field, the well-known Vickrey-Clarke-Groves (VCG) mechanism is known to be incentive compatible and efficient in terms of users' utilities, yet it is computationally hard to use it in real systems. As a result, many real world implementations use various approximations of VCG [5], [8]–[10] thus compromising incentive compatibility in its game-theoretical meaning.

This paper is based on a previous work which explored the effects of preemption and migration on fairness and performance in clusters [11]. This work presented a design of a preemptive job scheduler, called Online Greedy Migration (GM) and performed extensive simulations using workloads from real clusters. The data that was missing in these workloads, e.g. job priorities, was randomly generated according to a bimodal distribution. The GM scheduler was oblivious of any economic aspects, such as bids and payments. As a result, user truthfulness and user utility issues were not dealt with. The authors have evaluated the performance by using the Bounded SlowDown (BSD) metric (defined in Section II) and the Severely Slowdowned Jobs (SSJ) metric (defined in Section III). The conclusion of this work was that preemptive scheduling improves the performance and the fairness.

This paper extends the work presented in [11] by including market considerations and exploring user truthfulness issues. The goal is to analyze the effects of users' untruthful behavior, suggest a way to discourage it and to explore the stability of the market. To this end, in Section II, we transform the GM scheduler into a market mechanism by allowing it to support various payment schemes. We also introduce the Small Risk-aggressive Group (SRG) untruthfulness model which provides a simple characterization of users' behavior. The study was performed by running real-world workload traces in a simulated environment. The rationale for performing simulations was to save time and resources, since normally workloads span months (or even years) and require clusters with hundreds of nodes. Although there is a significant amount of theoretical work in the field of Algorithmic Mechanism Design [9], [12], [13], these results do not generalize to our much more complicated setting due to: (a) an *online* environment in which the number and the arrival time of future jobs is unknown; (b) no job run-time information is available upon job's submission; (c) the preemption capability of the scheduler. In Section III we show that untruthful users experience degraded performance in that their slowdown is higher than for truthful users. This section also verifies that compared to other non-preemptive scheduling algorithms, our upgraded scheduler retains the same properties as the GM scheduler. The main contribution of this paper, described in Section IV, is that the $k$-th price payment scheme discourages untruthful users from attempting to game the market. In addition to being robust, it also increases the utility of all the users compared to the first price payment scheme. These results show the advantage of preemptive migration in clusters because it enables to use the $k$-th price scheme in our mechanism. Section V explores the stability of the market by formulating a one-shot game between the mechanism designer and untruthful users. It is shown that there are specific combinations of strategies that produce equilibria in this game. Finally, Section VI discusses related work and Section VII presents the conclusions and ideas for future work.

## II. Computing Market Models

This section presents the basic computing and the untruthfulness market models.

### A. The Basic Computing Model

Consider a set of $n$ homogeneous (same architecture and same speed) computers (a cluster or a multi-cluster) in which an online bidding scheduling algorithm is used to assign incoming sequential jobs to computers (nodes). Assume that each node runs only one job at a time and that the system supports job preemption; i.e., a job can be stopped at any stage of its execution and be resumed later, not necessarily in the same node. Before starting a job, each user must determine his or her private value, which is the maximal amount that the user is willing to pay per unit of run-time. Note that the private value is not known to the other users. Upon submitting a job each user states a bid, called the reported bid, which might be lower than the private value. After the job is submitted the bid cannot be changed. Throughout this article the notions of jobs and users are used interchangeably.

The following Highest-Bid (HB) online algorithm is based on the Online Greedy Migration (GM) algorithm presented in [11], with the addition of a payment calculation. The algorithm is used to determine whether to assign a job to a node or to place it in a queue of waiting jobs. It supports various payment schemes by assuming that $p'_i$, which is the current payment per unit of run-time for job $i$, was already calculated somehow by one of the payment schemes.

In terms of algorithmic mechanism design, users has well defined *utility functions* (see below) that represent their preference over various outcomes of the Highest-Bid algorithm. It is assumed that users try to game the system, for example by reporting false information to optimize their respective utilities [7].

**Algorithm 1** Highest-Bid (*HB*)

Upon job arrival, job termination or bid update do:

1) If it is a new job $j$ set its total payment: $p_j = 0$.
2) For each running or newly terminated job $i$, set: $p_i = p_i + p'_i t'_i$, where $t'_i$ is the elapsed time since the last assignment of the job and $p'_i$ is the current payment per unit of run-time.
3) Sort the set of new and currently queued and uncompleted jobs in a descending order according to their bid. Break ties according to the submission time.
4) Assign the first $n$ jobs from the sorted list to the $n$ nodes, possibly preempting jobs with lower bids.
5) For each assigned job $i$, determine its current $p'_i$, and set: $t'_i = 0$.
6) Queue unassigned jobs until the next run of the algorithm.

For each job $i$, let:

- $w_i$ and $\tilde{w}_i$ be the private value and the reported bid.
- $\varphi_i$ be the flow time (in seconds), defined as the total time between the submission time and the time when the job was finished. Note that $\varphi_i$ depends on $\tilde{w}_i$.
- The job's Bounded Slow-Down ($BSD_i$) factor be:

$$BSD_i = \varphi_i / \max\{\tau_i, 60\}, \tag{1}$$

  where $\tau_i$ is the actual run-time in seconds.

Note that the traditional slowdown metric is defined by: $\varphi_i/\tau_i$. However, having a few seconds (short) jobs and long jobs in the same workload might produce excessive values of slowdown for the short jobs. Hence we use a special slowdown metric which bounds (from below) the run-time of a job to 60 seconds, which is best suited for our workloads.

Following the definition in [14], the user's valuation $v_i$ of job $i$, which represents the user's preference for a shorter completion time, is defined by:

$$v_i = -w_i \varphi_i. \tag{2}$$

**Definition: *Quasi-Linear Utility***
*The Quasi-Linear Utility function $u_i$ of job $i$ is defined by: $u_i = v_i - p_i$.*

Using Eq. 2, the utility function of job $i$ becomes:

$$u_i = -w_i \varphi_i - p_i. \tag{3}$$

Ideally users truthfully reveal their private values. This can formally be defined by:

**Definition: *Full-Truthfulness (FT) Model***
*For each job $i$: $w_i = \tilde{w}_i$.*

However, since users are *self-interested*, *rational* and *strategic*, they may report untruthful bids, i.e. $w_i \neq \tilde{w}_i$ if it maximizes their utility.

Based on the analysis in [3], it is assumed that the distribution of the private values is bimodal which is defined as a statistical distribution having two modes. Fig. 1 presents the distribution that we use: 80% of the values come from a low normal distribution (less important jobs) with mean of 30, and the remaining 20% of the values come from a high normal distribution (more important jobs) with mean of 150. Both of these distributions have a standard deviation of 15. Throughout this paper, *low-bid jobs* refer to jobs with $w_i \in [0, 60]$, *middle-bid jobs* refer to jobs with $w_i \in (60, 120)$ and *high-bid jobs* refer to jobs with $w_i \in [120, 180]$. This division provides a better insight into the results.

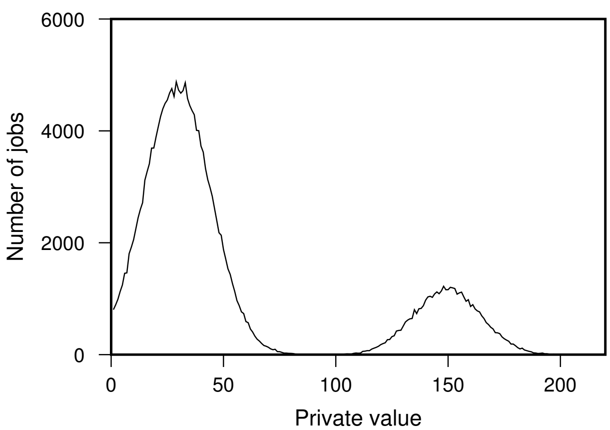



Figure 1. **Bimodal distribution of private values**

### B. The Untruthfulness Model

We now define a model for an untruthful behavior of users in the computing model defined in the previous section.

Let the reported bid range be defined by:

$$R = [(1 - \beta)w_i, w_i], \tag{4}$$

where $0 \leq \beta < 1$. Assume that users strongly prefer to avoid higher payments. Assume that $\tilde{w}_i \in R$ is chosen randomly and uniformly from the range $R$. We now define the users' untruthful behavior model used in this paper:

**Definition: *Small Risk-aggressive Group (SRG) Model***
*Divide all of the jobs randomly and uniformly into two groups: the first, called "risk-conservative", contains 90% of the jobs with* $\beta = 10\%$. *The second group, called "risk-aggressive", contains the remaining 10% of the jobs with* $\beta = 90\%$.

Informally, our first assumption in the SRG model is that users facing uncertainties have simple strategies, and therefore can only underbid their values. Specifically, users would never overbid and would never delay their arrival time to the system. Our second assumption is that the majority of users are conservative. Therefore, their willingness to underbid is very small while only a small group of aggressive users can severely underbid their values.

Although the SRG model assumes that all of the jobs are known in advance, it also can be used without knowing the total number of jobs. Each time a new job is submitted, a biased 10-90 coin is flipped so that the job has a 10% chance of being "aggressive" and a 90% chance of being "conservative". More elaborated models of user untruthfulness could make the market more complicated to analyze and alter the original distribution of the private values. For example, increasing the percentage of the risk-aggressive users to 80% and lowering the percentage of the risk-conservative users to 20% (with the same values of $\beta$) results in a more exponential rather than bimodal distribution. We note that in SRG, the distribution of the bids is nearly identical to that of FT. Also, note that the distribution of $\tilde{w}_i$ is smooth since for both groups (risk-conservative and risk-aggressive) it is chosen uniformly and randomly from the range $R$.

We note that we also investigated a more traditional division of 80% of the jobs in the risk-conservative group and the remaining jobs in the risk-aggressive group, with the $\beta$ parameter set to 20% and 80% respectively. The results turned out to be nearly identical to that of the SRG division, thus we studied only the latter.

## III. Performance of the Highest-Bid with SRG

In this section we compare the performance of Highest-Bid preemptive algorithm with the Full-Truthfulness (FT) and the Small Risk-aggressive Group (SRG) models. We also compare this algorithm to other, non-preemptive scheduling algorithms.

### A. Full-Truthfulness vs. SRG

We simulated the Highest-Bid algorithm using real-world workloads from three homogeneous clusters [15]: DAS2 [16], LPC [17] and REQUIN [18]. Each workload consists of information about each job, including the submission time, the actual run-time, the user's estimate of the run-time and other, non-relevant information. The private values for the jobs were generated using the bimodal distribution described in Section II.

For each workload, Table I presents the number of nodes used (in the simulations), the number of jobs and the mean (average) run-time of all the jobs (in seconds). Note that since our model does not deal with parallel jobs, we converted such jobs to sequences of serial jobs. Also note that due to under-utilization of both the DAS2 and the LPC clusters, we artificially reduced the number of nodes in these clusters by 25%, in order to create a competitive environment. Although the simulations were performed for all of the above workloads, we present only the results of the LPC workload, since the results for the other workloads were nearly identical.

Table I
CLUSTERS AND WORKLOADS

| Cluster name | DAS2 | LPC | REQUIN |
|---|---|---|---|
| Number of nodes | 48 | 105 | 1,536 |
| Number of serial jobs | 118,567 | 219,704 | 466,177 |
| Average run-times | 2,236 | 3,212 | 45,674 |

*1) Average Bounded Slowdown vs. Private Value:* The average bounded slowdown for different private values $w_i$ is presented in Fig. 2. In the figure, the dashed line represents the risk-aggressive users in the Small Risk-aggressive Group model, while the solid line represents all of the users in the Full-Truthfulness model.

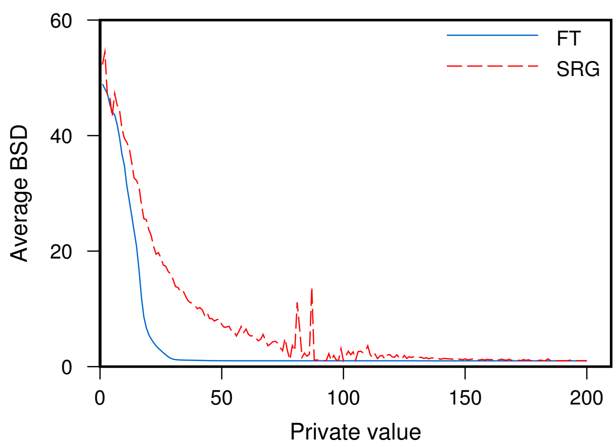



Figure 2. **Bounded slowdown comparison**

From the figure it can be seen that the average bounded slowdown of the low-bid jobs is higher in the Small Risk-aggressive Group model. The number of middle-bid jobs is small, which causes noisy results for this range of bids. Although for high-bid jobs there is almost no difference between the models, a more careful comparison below indicates that the slowdown in the case of Small Risk-aggressive Group model is higher. Since the Highest-Bid algorithm prefers jobs with higher bids over jobs with lower bids, lowering a job's bid increases its slowdown. As a result, risk-aggressive jobs are subjected to a greater slowdown.

*2) Measure of Severely Slowdowned Jobs:* Based on the analysis in [3], it is assumed that users show tolerance to the growing slowdown up to a certain threshold. We now define this threshold value for excessive bounded slowdown values:

**Definition: *Severely Slowdowned Job (SSJ)***
*A job $i$ such that $BSD_i \geq 5$.*

We note that the value of 5 was chosen based on the analysis in [11].

Fig. 3 presents the percentage of SSJs for different private values $w_i$. In the figure, the dashed line represents the risk-aggressive users in the Small Risk-aggressive Group model, while the solid line represents all of the users in the Full-Truthfulness model.

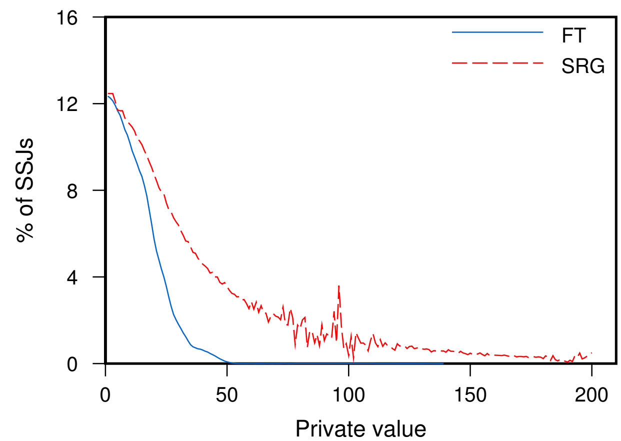



Figure 3. **Severely slowdowned jobs comparison**

From the figure it can be seen that the percentage of low-bid SSJs is higher in the Small Risk-aggressive Group model than in the Full-Truthfulness model. As before, the results for middle-bid jobs are noisy. This means that a non-negligible percentage of risk-aggressive users' jobs are subject to excessive slowdown. It is assumed that a job's bid correlates with its importance and that the user wants more important jobs to have less slowdown. Therefore, good performance means that jobs with higher bids do not become SSJs and have the lowest slowdown possible. The above results show that risk-aggressive users are penalized by degraded performance compared to the performance they would have received had they reported truthful bids.

### *B. Highest-Bid vs. Non-Preemptive Algorithms*

In this section we compare the performance of the Highest-Bid preemptive algorithm to several non-preemptive algorithms using the adjusted workloads described in the previous section.

The following three algorithms were used in the analysis:

1) NPHB - a non-preemptive version of the Highest-Bid algorithm that sorts the jobs in the queue according to their reported bid and then assigns jobs to available nodes.
2) WSPT - similar to NPHB, except that the queue is sorted according to the ratio between the reported bid and the run-time: $\tilde{w}_i/\tau_i$, known as the Smith prominent ratio rule [19]. Note that this algorithm assumes that the run-times are known.
3) WSEPT - a version of WSPT, in which the actual run-time $\tau_i$ is replaced by a user estimation of the run-time

$\tilde{\tau}_i$. Note that the run-time estimations either appear in the workloads of the DAS2 and the LPC cluster or were generated for the REQUIN cluster based on [20].

*1) Average Bounded Slowdown vs. Private Value:* The average bounded slowdown of the risk-aggressive users for different private values $w_i$ and for the above three algorithms is presented in Fig. 4. In the figure, the solid line represents the Highest-Bid algorithm, while the rest of the lines represent the other algorithms.

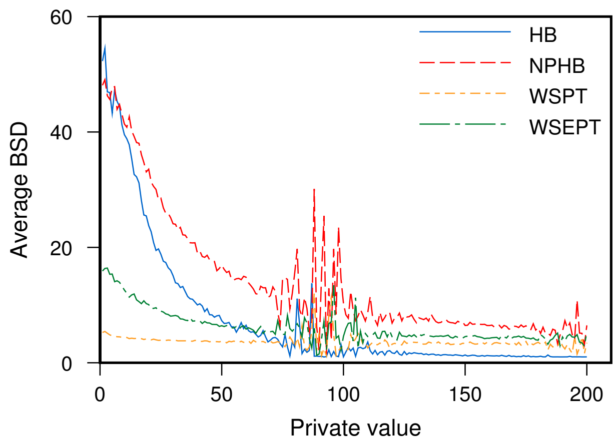



Figure 4. **Bounded slowdown comparison**

From the figure it can be seen that WSPT performs better than WSEPT, and that both of these algorithms perform better than HB for low-bid jobs. Observe that for high-bid jobs, the performance improves and the average bounded slowdown becomes smaller. As before, the results for the middle-bid jobs are noisy. Also, the HB algorithm outperforms the NPHB algorithm for all bids. We note that the last result confirms the results presented in [11]; except that in our case the result provides a new perspective about the Small Risk-aggressive Group model.

*2) Measure of Severely Slowdowned Jobs:* The percentage of SSJs of the risk-aggressive users for different private values $w_i$ for the HB (solid line), NPHB, WSPT and the WSEPT algorithms is presented in Fig. 5. From the figure it can be seen that the HB algorithm has consistently fewer SSJs for almost all of the bids, which means that its performance is better than that of the other algorithms. As in the previous case, the results confirm those of [11].

## IV. Analysis of Different Payment Schemes

In the previous section it was shown that risk-aggressive users that use the Highest-Bid algorithm are penalized in terms of the bounded slowdown for reporting untruthful bids. This section provides an analysis of users' utilities under the "first price" and the "k-th price" payment schemes (defined below), for both the Small Risk-aggressive Group and the Full-Truthfulness models. As in the previous section, only

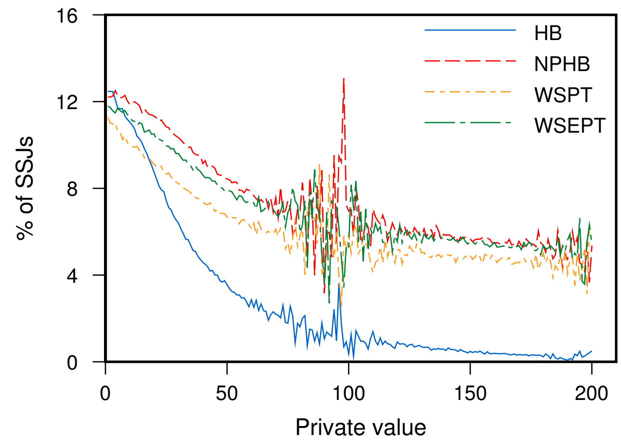



Figure 5. **Severely slowdowned jobs comparison**

the results of the LPC workload are presented since the results for the other workloads were conceptually the same.

**Definition:** ***First price***

*Each running job pays exactly the bid it reported.*

In the first price payment scheme, whenever the number of jobs is smaller than the number of nodes, the payment of each job becomes the reservation price of the node. Obviously, it is economically unreasonable to request a full price if there is no demand. Note that in order to simplify the model, the reservation price of each node is set to 1.

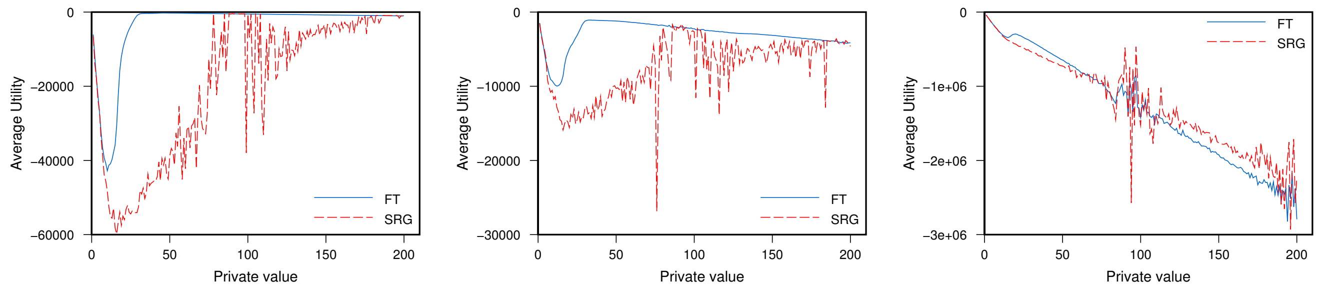
**Definition:** ***k-th price***

*Each running job pays the bid of the job with the highest bid in the queue; if the queue is empty, the payment of each job becomes the reservation price of the node.*

Note that the $k$-th price payment scheme is a dynamic extension of the classical second price scheme (the price is recalculated every time a job is assigned to a node). Essentially, in this scheme, each job pays the minimal bid it could have reported and still be running.

The utility function (Eq. 3) becomes linearly lower when the run-time of a job is increased; i.e., comparing the utility value of a short job with the utility value of a long job almost always shows that the short job has a higher utility. This effect is reduced when dividing the jobs according to their run-times and comparing the corresponding utility values within the same run-time group. Therefore, in the analysis that follows, all of the jobs were divided into three equally sized groups: $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ according to their respective run-times. Note that this division is conceptually different from the division into low-bid, medium-bid and high-bid sub-ranges.

Fig. 6 presents the average utility values of the first price scheme for different private values $w_i$. In the figure, the dashed line represents the risk-aggressive users in the Small Risk-aggressive Group model, while the solid line represents




(a) Group $\mathcal{A}$ (b) Group $\mathcal{B}$ (c) Group $\mathcal{C}$

Figure 6. **Utility comparison of jobs using the first price scheme**

all the users in the Full-Truthfulness model. From the figure it can be seen that for groups $\mathcal{A}$ and $\mathcal{B}$, the utility values of both low-bid and high-bid jobs are higher for the Full-Truthfulness model. Intuitively, this effect is more significant for low-bid jobs; since there is a bigger amount of such jobs in our bimodal distribution, under-bidding causes huge competition and delays. In the case of group $\mathcal{C}$ (Fig. 6(c)), the utility values of the low-bid jobs are lower in the Small Risk-aggressive Group model than in the Full-Truthfulness model. Note that for high-bid jobs this phenomenon is reversed. Intuitively, since there is a small amount of high-bid jobs and since these jobs are long running, it is beneficial to underbid as the competition among these jobs diminishes over time. This is further shown in Fig. 7(a) and Fig. 7(b).

From Fig. 2 it can be seen that for the Small Risk-aggressive Group model the bounded slowdown values of high-bid jobs are very small and essentially coincide with the values of the Full-Truthfulness model. Thus by paying less, the risk-aggressive users benefit in terms of the overall utility. This result is not surprising since the first price payment scheme inherently lacks incentive compatibility. However, from the mechanism designer perspective, this is a problematic situation since risk-aggressive users are encouraged to game the system whereas risk-conservative users are left at the mercy of the aggressive users. The development of a formal, fully incentive compatible *online* mechanism for our model is beyond the scope of this paper. Therefore, a first step towards incentive compatibility in our model can be made by using the $k$-th price payment scheme, which is known to be incentive compatible in auctions with multiple winners [7]. The job assignment step in the Highest-Bid algorithm which assigns $n$ jobs from the queue to $n$ nodes, can be viewed as a momentary auction with $n$ winners. Thus the $k$-th price scheme is an approximation of a fully incentive compatible mechanism in our case.

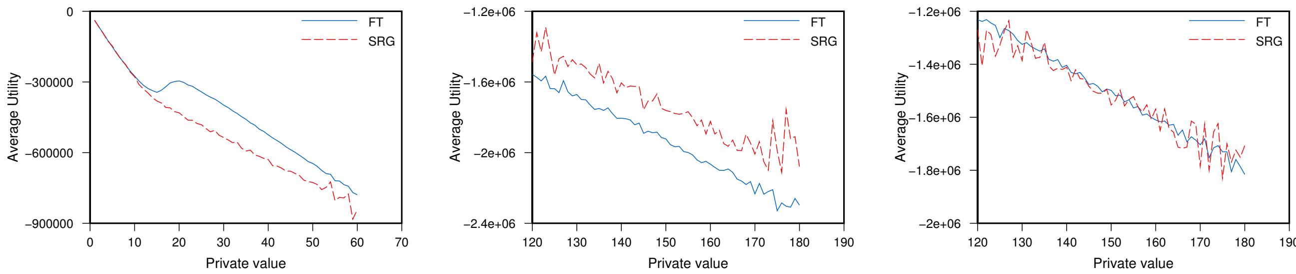
Fig. 7(c) presents the utility values of high-bid jobs using the $k$-th price scheme. From the figure it can be seen that the utility values of the risk-aggressive users in the Small Risk-aggressive Group model coincide with the utility values of the users in the Full-Truthfulness model. By comparing the results in Fig. 7(b) to Fig. 7(c) it can be seen that there is no incentive for risk-aggressive users to report smaller bids since the utility would remain essentially the same as if they had reported truthful bids.

In order to understand the differences between the first price and the $k$-th price schemes we compared the utility values in these cases for both of the truthfulness models. It turns out that the utility values are the same for group $\mathcal{A}$ and almost the same for group $\mathcal{B}$, with $k$-th price having slightly higher values. The most significant differences are presented in Fig. 8 which compares the utility values of high-bid jobs from group $\mathcal{C}$. The figure shows that the utility values of the $k$-th price are higher for both of the models. In this scheme each payment is no greater than a job's reported bid, which is exactly the payment in case of the first price. By Eq. 3, the utility value for the $k$-th price scheme is not lower than the utility value for the same job in the case of the first price scheme. This also explains the results in Fig. 7(c) in which the utility values of the $k$-th price scheme in the Full-Truthfulness model become higher and virtually equal to the utility values of the risk-aggressive users.

The conclusion of this section, which is also the main contribution of this paper is that risk-aggressive users can benefit from reporting untruthful bids under the first price payment scheme. However, using the $k$-th price scheme, which is more robust than the first price scheme, results in higher utility values for all types of users and also prevents risk-aggressive users from deriving any benefit from reporting untruthful bids. Although the $k$-th price scheme might seem harder to implement, it is actually simpler than other VCG-like mechanisms [5]. We note that although similar results about the $k$-th price scheme are known in auctions theory [7], our work is novel in the context of online and preemptive computing model.

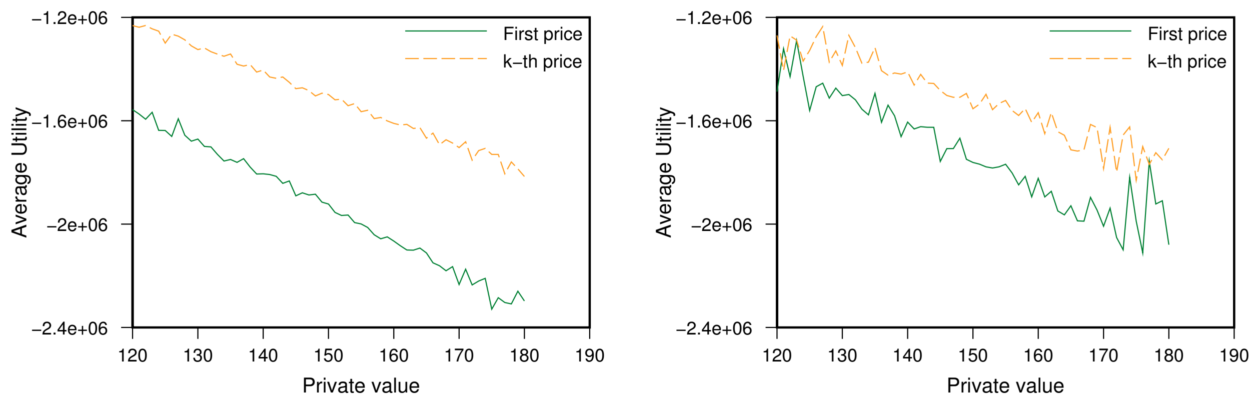
## V. MARKET STABILITY

This section presents an empirical study of market stability in the context of online computational markets. We analyze market stability in terms of risk-aggressive users' utility and the mechanism designer's revenue. Since stability is




(a) Low-bid jobs (first price)
(b) High-bid jobs (first price)
(c) High-bid jobs ($k$-th price)

Figure 7. **Detailed utility comparison of jobs in group $\mathcal{C}$**




(a) High-bid jobs (FT)
(b) High-bid jobs (SRG)

Figure 8. **Detailed utility comparison of jobs in group $\mathcal{C}$**

associated with formal theoretical equilibrium concepts, we formulate a simple one-shot game to model the interaction between the mechanism designer and risk-aggressive users. We then show that there exists a *unique pure (symmetric) Nash-equilibrium* in this game.

We first identify two major players: the mechanism designer and an aggregate consumer playing on behalf of the risk-aggressive users, where the former essentially wishes to maximize revenue and the latter strives to maximize personal (aggregated) utility. More formally, the strategy space of the mechanism designer is any combination of the possible allocation algorithms (HB, NPHB or WSPT) and payment schemes (first price or $k$-th price). Ideally, the mechanism designer wishes to maximize its revenue. However, since in real-life, consumers have several optional competing suppliers, the mechanism designer also needs to consider the quality of the service. Therefore, in our game we model the payoff of the mechanism designer in terms of the Total Weighted Flow Time (TWFT) studied in [11], as well. The strategy space of the risk-aggressive users is any of the following $\beta$-parameter values: $5\%, 25\%, 50\%, 60\%, 70\%, 80\%, 90\%$. Note that the specific value of $\beta$ is chosen only once before the submission of any job and cannot be changed later.

Since the NPHB and the WSPT algorithms lack the ability to preempt jobs and since the queue of the waiting jobs in WSPT has a different order, the $k$-th price scheme had to be modified. Specifically, in NPHB, the $k$-th price was reduced to the first price whenever the bid of a running job was smaller than the maximal bid of a queued job. Also, in WSPT, the price for a running job $i$ was set to $(\tilde{w}_j/\tau_j)\tau_i$, where $\tilde{w}_j/\tau_j$ is the largest ratio among the queued jobs. This last price is the minimal bid that job $i$ could have reported and would still be scheduled to run.

Table II and Table III present the payoff matrices of the game for the LPC workload, for the first price payment scheme and the $k$-th price payment scheme respectively. As in the previous section, both tables present only the results for group $\mathcal{C}$, since the corresponding results for groups $\mathcal{A}$ and $\mathcal{B}$ are nearly identical. Note that the results for the DAS2 and REQUIN workloads were nearly identical to the LPC workload and thus are omitted.

As in the standard payoff matrix, the rows in both Table II and Table III represent the strategies of the mechanism designer, whereas the columns represent the risk-aggressive users. Each cell in the matrix contains two payoff values, the left is for the mechanism designer and the right is for the risk-aggressive users.

Table II
PAYOFF MATRIX FOR THE FIRST PRICE SCHEME (GROUP $\mathcal{C}$)

| Algorithm | 5% | 25% | 50% | 60% | 70% | 80% | 90% |
|---|---|---|---|---|---|---|---|
| HB | **1.000**, 0.833 | **0.990**, 0.842 | **0.976**, **0.845** | **0.970**, 0.838 | **0.963**, 0.824 | **0.955**, 0.802 | **0.945**, 0.777 |
| NPHB | 0.862, 0.801 | 0.853, 0.810 | 0.841, **0.813** | 0.836, 0.809 | 0.831, 0.799 | 0.824, 0.785 | 0.816, 0.767 |
| WSPT | 0.904, 0.827 | 0.895, 0.842 | 0.883, 0.860 | 0.878, 0.863 | 0.873, 0.863 | 0.868, **0.867** | 0.863, 0.861 |

Table III
PAYOFF MATRIX FOR THE $k$-TH PRICE SCHEME (GROUP $\mathcal{C}$)

| Algorithm | 5% | 25% | 50% | 60% | 70% | 80% | 90% |
|---|---|---|---|---|---|---|---|
| HB | 0.306, **1.000** | 0.303, 0.988 | 0.296, 0.961 | 0.292, 0.940 | 0.287, 0.913 | 0.281, 0.882 | 0.274, 0.846 |
| NPHB | 0.636, 0.854 | 0.629, **0.858** | 0.619, 0.847 | 0.614, 0.839 | 0.609, 0.827 | 0.603, 0.810 | 0.597, 0.790 |
| WSPT | **0.774**, 0.855 | **0.765**, 0.869 | **0.755**, 0.879 | **0.751**, 0.884 | **0.746**, **0.886** | **0.741**, 0.881 | **0.736**, 0.879 |

The payoffs of the mechanism designer are values of the revenue and quality of service metric, defined by: $\sum_i p_i / \sum_j w_j \varphi_j$, where $i, j$ are for all jobs. Note that the numerator is the sum of the payments of all the jobs, while the denominator is the TWFT of all the jobs. Also note that the payoff has a standard economical meaning since the TWFT reflects the total social welfare (it is the negated sum of user valuations defined by Eq. 2). In the tables, each value is normalized by the maximal payoff (among all the values of both tables), which is the payoff for the combination of Highest-Bid with $\beta = 5\%$ with the first price scheme. The payoffs of the risk-aggressive users are their average utility values normalized by the maximal utility value (among all the values of both tables), which is the payoff for the combination of Highest-Bid with $\beta = 5\%$ with the $k$-th price scheme. In the tables, the maximal payoffs of various strategies played by both players are shown in bold.

The Nash equilibrium for the first price scheme occurs for the combination of HB with $\beta = 50\%$. Observe that in this case the payoff of the mechanism designer is 0.976 and is very close to the optimum 1.000. For the $k$-th price scheme, the Nash equilibrium occurs for the combination of WSPT with $\beta = 70\%$.

The total *revenue* in case of the first price payment scheme is independent of the actual algorithm. Therefore, the payoff values are effectively reduced into weighted flow time values. Thus, since the Highest-Bid algorithm has the best performance in terms of weighted flow time, it has the maximal payoff values of the mechanism designer, who would prefer to combine it with the first price scheme (which gives higher revenue). However, this choice is problematic from the users' perspective because it does not provide the highest possible utility and motivates them to game the mechanism. Had our mechanism been a monopoly and if users had not had any alternative, the aforementioned problems could have been ignored. However the prevalence of other clusters and allocation mechanisms drives users to seek other mechanisms that provide better utility and better stability.

Table III shows the details of the payoffs of the $k$-th price scheme. From the table it can be seen that the Nash equilibrium occurs for the combination of WSPT with $\beta = 70\%$. This last algorithm has the advantage of knowing the run-times of the jobs, which on one hand allows the mechanism designer to get better payoffs, but at the same time it does not conform with our model. Examining the results without this algorithm produces an equilibrium for the combination of NPHB with $\beta = 25\%$. Although in terms of weighted flow time the Highest-Bid algorithm is better, the payoffs from the NPHB algorithm are higher because of its higher revenues. Note that Table III strengthens our results from Section IV: by using the Highest-Bid algorithm with the $k$-th price scheme, the utility of the users is indeed maximized when the reported bids are truthful ($\beta = 5\%$).

To conclude, there is no clear recommendation as to which combination of algorithm and payment scheme to use. By modeling the interaction between the mechanism designer and the risk-aggressive users as a one-shot game, we were able to analyze the market stability in our model. The results indicate that there are Nash equilibria for both the first price and the $k$-th price schemes. However, in both cases the equilibria occur when risk-aggressive users report untruthful bids. Clearly, the combination of the HB algorithm with $k$-th price discourages these users from reporting untruthful bids since it reduces their utility. Further studies of market stability are necessary to understand how user behavior is affected and to determine whether an equilibrium in which users are truthful can be achieved in our or some closely related models.

## VI. RELATED WORK

Market-based resource allocation in clusters, multi-cluster grids and clouds has been extensively studied [1], [2], [4], [5], [12], [21]. The two main approaches are: the *spot-market* and the *reservation-market*. Spot-markets use online auctions with an unknown number of users that can submit jobs anytime. On the other hand, reservation-markets use periodic auctions that gather bids and then clear the market at regular intervals by determining the winners and the payments. In this last approach resources are reserved for the winners of the previous auction until the next auction. The reservation-market approach is commonly used with batch schedulers, which require additional information such as the job run-times.

Tycoon is an example of a resource allocation system that consists of a number of components, including resources, agents, a resource discovery service and a trusted bank [4]. The resource discovery service is used to get the latest state of each node, such that users can bid separately on each node. Nodes are independent and each one runs multiple jobs simultaneously in proportion to the job's bid. The agent's role is to optimize, on behalf of the user, the task of finding the nodes in which the job could get a larger share for a smaller price. Each user pays the bid of the job directly unless there is no load on the node, in which case only the reservation price is paid.

A theoretical study of a job scheduler in a cluster environment is presented in [12]. The authors formulate an online model with $k$ resources and continuously arriving jobs that require resources. Each request for resources consists of the bid, the length of the job, the arrival and the departure time of the job. An auction takes place each time a new job arrives or a running job terminates. As in many "single-parameter" auctions, each job pays the amount of the smallest bid that it could have reported and still get an allocation. If a job does not finish running before the departure time, the utility is set to be zero and the job pays nothing. By assuming restricted misreporting of the arrival and departure times and that all job lengths are in the $[L_{min}, L_{max}]$ interval, it is shown that the allocation mechanism is incentive compatible and is $O(log(L_{max}/L_{min}))$-competitive compared to the offline case [12]. The work presented in this paper uses a greedy allocation and a payment scheme ($k$-th price) which resembles the approach used in [12]. The difference, however, is that we do not require the departure times of the jobs and the job lengths.

Incentive compatibility is a desired property in auctions since it encourages users to truthfully reveal their information, be it a job bid or job run-time. Achieving incentive compatibility in combinatorial auctions is computationally infeasible (NP-complete [7]). Significant efforts [5], [8]–[10] have been made to devise various approximate algorithms that provide restricted incentive compatibility. In [9] the authors introduce a greedy allocation scheme that ensures incentive compatibility for a restricted type of users. The authors of [5] implemented a reservation-market mechanism that restricts the expressiveness of users' requests for resources. By using dynamic programming the authors presented a computationally tractable incentive compatible mechanism. This mechanism was evaluated by modeling the untruthful behavior of a single user by randomly varying the bid. In the current paper we used the last approach with a more elaborate model in which there is a randomly selected group of untruthful users.

The work of Ng et al. [22] analyzes user behavior in a real working market-based resource allocation system, called Mirage. This system allocates testbed resources in repeated first price combinatorial auctions [2]. Since it is inherently not incentive compatible, Mirage makes it possible to study users' behavior and the strategies employed to game the mechanism. Four strategies are described, the first of which is the strategy of lowering the reported bid and which we used in our SRG model.

## VII. Conclusions and Future Work

In this work we studied an *online* model of a computing market that effectively sells computing nodes in a cluster (or multi-clusters) to newly arriving jobs. In addition, we used a simple model of user behavior (the Small Risk-aggressive Group model) which assumes that the majority of the users are *risk-conservative* while a small minority are *risk-aggressive*. By using the greedy-preemptive Highest-Bid algorithm it was shown that although untruthful risk-aggressive users are penalized for their behavior in terms of the slowdown, they are still able to increase their utility if the first price payment scheme is used. Switching to the $k$-th price scheme resulted in lower utility for the risk-aggressive users; in fact, the utility values were effectively the same as in the case of full truthfulness. The main contribution of this paper is that in the context of our model which uses an online preemptive algorithm that does not assume knowledge of the run-times, choosing the $k$-th price scheme discourages users from attempting to cheat the mechanism.

In this work we also analyzed the market stability from the point of view of a two-player game. Although we discovered that there are strategy combinations that yield equilibria, no clear recommendation about the stability in the market can be made. Our results indicate that stability does not necessarily mean full truthfulness. Further work is necessary to study this type of game between the users and the mechanism designer.

Our experimental work can be extended in several ways. First, it would be interesting to conduct a theoretical study of the models presented in this work. Another interesting problem is to study the stability issues with more than two players, e.g., with many competing mechanism designers and more choices of strategies. Yet another direction would be to study different budget models and to analyze the effects of limited budgets on the performance and utility.

## Acknowledgments

We wish to thank Dror Feitelson, John Ledyard, Stuart McDonald and Noam Nisan for helpful discussions.

This work was supported in part by the EU IST program, grant 034286 "SORMA".

## References


[1] R. Buyya, C. S. Yeo, and S. Venugopal, "Market-oriented Cloud computing: Vision, hype, and reality for delivering it services as computing utilities," in *Proc. 10th IEEE Int'l Conf. on High Performance Computing and Communications (HPCC)*, Sep. 2008, pp. 5–13.

[2] B. N. Chun, P. Buonadonna, A. AuYoung, C. Ng, D. C. Parkes, J. Shneidman, A. C. Snoeren, and A. Vahdat, "Mirage: a microeconomic resource allocation system for sensornet testbeds," in *Proc. 2nd IEEE Workshop on Embedded Networked Sensors (EmNets)*, May 2005, pp. 19–28.

[3] B. N. Chun and D. E. Culler, "User-centric performance analysis of market-based cluster batch schedulers," in *Proc. 2nd IEEE/ACM Int'l Symposium on Cluster Computing and the Grid (CCGrid)*, May 2002, pp. 30–38.

[4] K. Lai, L. Rasmusson, E. Adar, L. Zhang, and B. A. Huberman, "Tycoon: An implementation of a distributed, market-based resource allocation system," *Multiagent Grid Systems*, vol. 1, no. 3, pp. 169–182, Aug. 2005.

[5] A. Mutz and R. Wolski, "Efficient auction-based grid reservations using dynamic programming," in *Proc. 2008 ACM/IEEE Conf. on Supercomputing (SC)*, Nov. 2008, pp. 1–8.

[6] R. Porter, "Mechanism design for online real-time scheduling," in *Proc. 5th ACM Conf. on Electronic Commerce (EC)*, June 2004, pp. 61–70.

[7] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic Game Theory*. Cambridge University Press, 2007.

[8] B. Edelman, M. Ostrovsky, and M. Schwarz, "Internet advertising and the generalized second price auction: Selling billions of dollars worth of keywords," *American Economic Review*, vol. 97, no. 1, pp. 242–259, March 2007.

[9] 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, Sep. 2002.

[10] A. Mutz, R. Wolski, and J. Brevik, "Eliciting honest value information in a batch-queue environment," in *Proc. 8th IEEE/ACM Int'l Conf. on Grid Computing (Grid)*, Sep. 2007, pp. 291–297.

[11] L. Amar, A. Mu'alem, and J. Stößer, "On the importance of migration for fairness in online grid markets," in *Proc. 9th IEEE/ACM Int'l Conf. on Grid Computing (Grid)*, Sep. 2008, pp. 65–74.

[12] M. T. Hajiaghayi, R. D. Kleinberg, M. Mahdian, and D. C. Parkes, "Online auctions with re-usable goods," in *Proc. 6th ACM Conf. on Electronic Commerce (EC)*, June 2005, pp. 165–174.

[13] R. Lavi and N. Nisan, "Online ascending auctions for gradually expiring items," in *Proc. 16th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Jan. 2005, pp. 1146–1155.

[14] B. Heydenreich, R. Müller, and M. Uetz, *Decentralization and Mechanism Design for Online Machine Scheduling*. LNCS, Springer, 2006, vol. 4059, pp. 136–147.

[15] D. Feitelson, "Parallel Workloads Archive," http://www.cs.huji.ac.il/labs/parallel/workload.

[16] "The Distributed ASCI Supercomputer 2 (DAS-2)," http://www.cs.vu.nl/das2.

[17] "Enabling Grids for E-science in Europe," http://www.eu-egee.org.

[18] "Shared Hierarchical Academic Research Computing Network," http://www.sharcnet.ca.

[19] W. Smith, "Various optimizers for single-stage production," *Naval Resource Logistics Quarterly*, vol. 3, pp. 59–66, 1956.

[20] D. Tsafrir, Y. Etsion, and D. Feitelson, *Job Scheduling Strategies for Parallel Processing*. LNCS, Springer, 2005, vol. 3834, pp. 1–35.

[21] A. Das and D. Grosu, "Combinatorial auction-based protocols for resource allocation in grids," in *Proc. 19th IEEE Int'l Parallel and Distributed Processing Symposium (IPDPS)*, April 2005.

[22] C. Ng, P. Buonadonna, B. N. Chun, A. C. Snoeren, and A. Vahdat, "Addressing strategic behavior in a deployed microeconomic resource allocator," in *Proc. 5th ACM SIGCOMM Workshop on Economics of Peer-to-Peer Systems (P2PECON)*, Aug. 2005, pp. 99–104.