Title: A Preemptive Truthful VMs Allocation Online Mechanism in Private Cloud

Author: Yonggen Gu Jie Tao Guoqiang Li Daniel W. Sun Xiaohong Wu Prem Prakash Jayaraman Rajiv Ranjan

PII:	S1877-7503(16)30083-7
DOI:	http://dx.doi.org/doi:10.1016/j.jocs.2016.05.006
Reference:	JOCS 503

To appear in:

Received date:	31-7-2015
Revised date:	10-12-2015
Accepted date:	11-5-2016

Please cite this article as: Yonggen Gu, Jie Tao, Guoqiang Li, Daniel W. Sun, Xiaohong Wu, Prem Prakash Jayaraman, Rajiv Ranjan, A Preemptive Truthful VMs Allocation Online Mechanism in Private Cloud, *<![CDATA[Journal of Computational Science]]>* (2016), http://dx.doi.org/10.1016/j.jocs.2016.05.006

A Preemptive Truthful VMs Allocation Online Mechanism in Private Cloud

Yonggen Gu[a], Jie Tao[a], Guoqiang Li[c], Daniel W. Sun[d,e], Xiaohong Wu[a,b], Prem Prakash Jayaraman[f], Rajiv Ranjan[g]

[a] *School of Information Engineering, Huzhou University, China*
[b] *Shanghai University of Finance and Economics, China*
[c] *School of Software, Shanghai Jiao Tong University, China*
[d] *Data61, CSIRO, Canberra, Australia*
[e] *School of Computer Science and Engineering, The University of New South Wales*
[f] *RMIT, Australia*
[g] *School of Computing Science, Newcastle University, United Kingdom*

Abstract

During the last decade, cloud-technology has presented considerable opportunities for high-performance computing(HPC). In addition, technical computing data centers have been able to maximize their return on investment(ROI). HPC system managers can leverage the benefits of a cloud model for their traditional HPC environments to improve scalability, simplify service access, accelerate collaboration or funding, enable pay-for-use, and improve efficiency. Many HPC clouds assume the form of private Infrastructure as a Service(IaaS). In practice, private cloud users may strategically misreport task values in order to achieve a high profit, and thus cloud providers cannot simply maximize the sum of allocated users' value, which is called social welfare. For this reason, designing a mechanism that reveals the truthful value of users with a concern for both random arrival tasks and maximizing social welfare is necessary. In this study, a model of an online mechanism for virtual machines allocation is built, a preemptive online mechanism is proposed, the truthfulness is proved, a competitive ratio is given, and several simulations are conducted using real tasks from a data center. The total values and completed tasks are compared to the online and offline allocations, respectively, according to different capacity. The simulations reveal that our mechanism is more efficient than the offline mechanism.

Keywords: Cloud computing, preemptive allocation, online algorithm, truthfulness, competitive ratio

Preprint submitted to Journal of Computational Science *December 6, 2015*

1. Introduction

Cloud computing has been employed by HPC and technical computing data centers to maximize return on investment(ROI). HPC system managers can leverage the benefits of a cloud model for their traditional HPC environments in order to improve scalability, simplify service access, accelerate collaboration or funding, enable pay-for-use, and improve efficiency. Many HPC clouds assume the form of private Infrastructure as a Service(IaaS). In private clouds, fixed price is easy to understand and primarily employed, but it is not economically efficient. By contrast, auction-based mechanisms are feasible solutions for virtual machines(VMs) allocation in clouds. An example of an auction-based VMs allocation mechanism is spot instances in Amazon EC2 [1]. This study focuses on auction-based mechanisms for cloud VMs allocation.

Most research has focused on resource allocation in offline situations, which do not consider the users' random arrival and departure at any time over a long period. Users ofen arrive randomly and must leave at a particular moment, which can be described as the users' lifetime. If sufficient cloud resources are allocated to the users in their lifetime, they can obtain a value that expresses payoff when a task is completed. This value is a user's private information. Our online mechanism is an appropriate response to the aforementioned situations. Designing a VMs allocation mechanism in a private cloud that maximizes the total value of the users is our primary challenge because of self-interested users and their random arrival.

In this study, we present a truthful preemptive VMs allocation online mechanism, and compare our mechanism with the optimal offline mechanism through experiments. Results reveal a good actual competitive ratio. In addition, we analyze the relationship between the performance and resource capacity. Our main contributions are the following: our online mechanism addresses the actual situation of a private cloud; truthfulness, which is the most important property, is satisfied; and the social welfare is high by the real data simulations.

This study is organized as follows. Related studies are described in Section 2. Section 3 describes our modeling of the preemptive VMs allocation online mechanism in cloud. Section 4 describes our online mechanism, for which we prove its truthfulness. In addition, allocation and payment algorithms are described. Section 5 describes the numerical experiments conducted to evaluate the performance of our online mechanism. We conclude this study in Section 6.

2. Related Work

A mechanism comprises a group of rules used for an aggregative outcome in which the participants are self-interested with the private information about their preferences. Mechanism design aims to identify the good system-wide means to obtain true participant preferences regarding outcomes. A mechanism is mainly used in the field of microeconomics and resource allocation. It is also used in distributed artificial intelligence and communication networks. In [2], a VCG-Kelly mechanism for allocating network resources was designed in which heterogeneous Internet traffic such as file transfers and real-time streams are regarded as flows. In [3], a mechanism for auctioning bundles of multiple divisible goods in a network was proposed and the weak Nash implementation for social-welfare maximizing allocation was proved. The study in [4] proposed a hierarchical auction model for network resource allocation in which the equivalent outcome was implemented and the efficient Nash equilibrium was proved. In [5], a double-sided auction market framework to address the challenges of decentralized complex node interactions was introduced. Its purpose was to employ bidding and charging strategies to maximize the social welfare.

Resource management and allocation have been considered a major problem in parallel and distributed systems for data-intensive applications, energy efficiency, massive simulations, and so on [7, 8, 9, 10]. The study in [6] considered that selfish users in a cloud environment may strategically compete for resources with others to maximize their own benefits. However, because this depresses overall system performance, a stochastic solution for allocation and payment outcomes was proposed to enforce cooperation and achieve efficient resource utilization in non-cooperative cloud systems. The study in [11] suggested three types of mechanisms: cloud-dominant strategy incentive compatible, cloud-Bayesian incentive compatible, and cloud optimal in order to prove or negate separately cloud resource procurement, incentive compatibility, budget balancing, and individual rationality. The study in [12] proposed a family of truthful greedy mechanisms for the allocation of bundles of VM instances, which is viewed as a multiple-units combinatorial auction, by formulating the VMs provisioning and allocation problem in clouds as an integer programming. Other studies on mechanism design of VMs provisioning in a cloud environment are provided in [13][14][15][16].

Some studies have examined cases in which resource allocation has been solved by means of online mechanisms. For examples, as plug-in hybrid electric vehicles have become more widespread, their charging must be coordinated because of electric grid capacity constraints. Model-free online mechanisms have been designed to guarantee truthfulness by occasional burning at each time step or departure [17][18]. In one study, a model-based precommitment mechanism was designed by modifying a well-known online algorithm consensus, and it achieved greater than 93% of the offline optimal [19]. Additional information about online mechanisms is provided in [20]. Some online mechanism frameworks applied to cloud computing have been built in [21][22]. In [23], a non-preemptive online mechanism for provisioning and allocating VM instances in clouds was proposed, which is incentive compatible with the M competitive ratio. Performance of the online mechanism was evaluated through experiments. This previous study considered the non-preemptive allocation situation only.

In this study, we consider the preemptive VMs allocation that is different from that examined in other studies. Many aspects of this type of VMs allocation are studied including the total value, completed tasks, actual competition ratio, and total payment. Simultaneously, a grouped mechanism is studied for additional payments.

3. VMs Allocation Online Mechanism Model

Providers of cloud computing provision VMs to users and aspire to maximizing revenue, utilizing resources, social welfare, and/or other objectives. Social welfare refers to the sum of value obtained by each user under a certain allocation. Private clouds are constructed by certain enterprises, institutions, and organizations. The users are limited in the internal members. Private clouds have the advantages such as rapid deployment and resources customization because the same principle applies to them as to public clouds. Compared to maximizing revenue in public clouds, maximizing social welfare is a reasonable goal in private clouds. In this study, we examine the VMs allocation mechanism, whose objective function is expressed as:

$$max \sum_i v_i \quad \forall i, \pi_i = 1 \tag{1}$$

where v_i is the value user i obtained if $\pi_i = 1$. The definition of π_i is given in equation (2) that follows. However, v_i denotes private information, which naturally cannot be revealed to cloud providers. Therefore, the mechanism must obtain v_i from users. A mechanism is truthful or incentive compatible if it can reveal the users' true private information.

Where the VMs are uniform in our model, C VMs exist in the cloud resource pool. Users come randomly and must leave at some pre-determined time. Tasks are single-valued, which means the user will hold v_i if a task is completed, otherwise he or she will hold a value of 0. Tasks are non-parallel and every task requires one VM. In other words, we treat a parallel task as a set of multiple independent non-parallel tasks. When one user arrives, he or she must tell the cloud provider his or her task information $\theta_i = (a_i, d_i, l_i, v_i)$, where a_i denotes the arrival time, d_i refers to the departure time, l_i denotes the task length, and v_i refers to the user's value if the task is completed before departure time. If the VMs allocation mechanism is truthful, the true θ_i will be reported. Otherwise, a strategic misreport may be given.

Time is divided into uniform period units and VMs are newly allocated at every unit. VMs allocation is preemptive, meaning that a single VM is held in unit t by user i but may be lost in $t+1$ if the task is not completed and it is prior to d_i. For example, when the unit length is 10 minutes, then $(100, 200, 50, 30)$ means that the arrival time is the 100^{th} unit, departure time is the 200^{th} unit, task length is 50 units, and the value is 30 if 50 units are allocated in the user's lifetime continuously or discretely.

The allocated units in user i's lifetime $[a_i, d_i]$ is expressed as l_i^a. The allocation is effective if $l_i^a = l_i$, whereas it is ineffective if $l_i^a < l_i$, which is described by the following.

$$\pi_i = \begin{cases} 1 & if \quad l_i^a = l_i \\ 0 & if \quad l_i^a < l_i \end{cases} \tag{2}$$

Certainly, superfluous allocation is not required. Therefore, $l_i^a > l_i$ is not considered. Note that π_i^t refers to user i's allocation at unit t. Thus, $\pi_i^t = 1$ means that a single VM is allocated to user i at unit t, and $\pi_i^t = 0$ means no VM is allocated.

We define $\theta_1 \prec \theta_2$ if satisfying

$$(a_1 \geq a_2) \cap (d_1 \leq d_2) \cap (v_1 \leq v_2) \cap (l_1 = l_2) \tag{3}$$

and at least one restrict $<$ or $>$ is requested. The allocation rule is monotonic if it satisfies

$$\pi_i(\theta_i, \theta_{-i}) = 1 \Rightarrow \pi_i(\theta'_i, \theta_{-i}) = 1 \quad \forall i, \theta_i \prec \theta'_i \tag{4}$$

4. Preemptive VMs Allocation Online Mechanism

4.1. Mechanism Description

A mechanism includes two rules: allocation and payment. Based on the VMs allocation online mechanism model, our mechanism M is given by the following rules:
Allocation rule: compute every effective user's priority value

$$g_i^t = \frac{v_i}{l_i - \lambda e_i^t}, \lambda \in [0, 1] \tag{5}$$

where e_i^t is the implemented length of user i's task up to t and λ is a constant between 0 to 1 in a certain mechanism. All g_i^t are sorted in descending order, the first C users are each allocated one VM respectively at unit t. If several g_i^t are equal, user i has a prior authority with the smaller a_i or earlier submission. Considering higher utility, if the user's remaining lifetime is less than the remainder task length, the user is ineffective.
Payment rule: the result is returned to the user at d_i and the user's payment is computed simultaneously. If the task is completed, the user pays a critical value v_i^c to the cloud providers. Otherwise, no result is returned and 0 should be paid. The critical value is defined as the lowest value that can guarantee completing the task prior to d_i if the user reports it to the provider.

4.2. Example

Considering $\lambda = 1$,$C = 1$, and three users, their types are $\theta_1 = (100, 104, 3, 30)$, $\theta_2 = (101, 104, 2, 25)$, and $\theta_3 = (102, 108, 3, 33)$. The allocation process is shown in Table 1, where Φ expresses the user who is ineffective and $allocation = 1$ in column 100 expresses that the VM is allocated to User 1 in time unit 100. $g_2^t = \phi$ in column 100 means User 2 is ineffective because his or her arrival time is 101. $g_1^t = \phi$ in column 103, because sufficient time units have been allocated to the User 1. Success or fail in the *result* column expresses whether l_i units are allocated to the corresponding user. The payment (critical value) of Users 2 and 3 is 22 for each. The analysis is shown in Tables 2 and 3. In Table 2, Users 2 and 3 maintain fixedness, allocation_1 occurs in $v_1 = 22 - \varepsilon$, and allocation_2 occurs in $v_1 = 22 + \varepsilon$, where ε is an arbitrarily small positive number. In Table 3, Users 1 and 3 maintain fixedness, allocation_1 occurs in $v_2 = 22 - \varepsilon$, and allocation_2 occurs in $v_2 = 22 + \varepsilon$. Table 2 shows that the task of User 1 will still be completed if he or she reports the value as $22 + \varepsilon$, but will not be completed if he or she reports the value as $22 - \varepsilon$. Therefore, the critical value of User 1 is 22. Table 3 shows the results of User 2 homoplastically.

Table 1: Allocation process

Time unit	100	101	102	103	104	Result
Allocation	1	1	1	2	2	
g_1^t	10	15	30	Φ	Φ	success
g_2^t	Φ	12.5	12.5	12.5	25	success
g_3^t	Φ	Φ	11	11	11	

Table 2: Payment (critical value) of User 1

Time unit	100	101	102	103	104	Result
Allocation_1	1	2	2	3	3	
$g_{1\_1}^t$	$\frac{22}{3} - \varepsilon$	$11 - \varepsilon$	$11 - \varepsilon$	$11 - \varepsilon$	$11 - \varepsilon$	fail
$g_{2\_1}^t$	Φ	12.5	25	Φ	Φ	
$g_{3\_1}^t$	Φ	Φ	11	11	22	
Allocation_2	1	2	2	1	1	
$g_{1\_2}^t$	$\frac{22}{3} + \varepsilon$	$11 + \varepsilon$	$11 + \varepsilon$	$11 + \varepsilon$	$22 + \varepsilon$	success
$g_{2\_2}^t$	Φ	12.5	25	Φ	Φ	
$g_{3\_2}^t$	Φ	Φ	11	11	11	

4.3. Truthfulness of Our Mechanism

For a single-value online mechanism, Lemma 1 has been proved in [15].

Table 3: Payment (critical value) of User 2

Time unit	100	101	102	103	104	Result
Allocation_1	1	1	1	3	3	
$g^t_{1\_1}$	10	15	30	Φ	Φ	
$g^t_{2\_1}$	Φ	12.5	12.5	$11-\varepsilon$	$11-\varepsilon$	fail
$g^t_{3\_1}$	Φ	Φ	11	11	16.5	
Allocation_2	1	1	1	2	2	
$g^t_{1\_2}$	10	15	30	Φ	Φ	
$g^t_{2\_2}$	Φ	12.5	12.5	$11+\varepsilon$	$22+\varepsilon$	success
$g^t_{3\_2}$	Φ	Φ	11	11	11	

Lemma 1: The mechanism is truthful if it satisfies the following conditions: (1) no early arrival and no late departures are misreported; (2) the known interesting set exists; (3) the deterministic monotonic allocation rule applies; and (4) the critical value is paid.

Theorem 1: The mechanism M is truthful.

Proof. No early arrival and no late departure misreports is obvious. Users can not attend to the mechanism in advance because programs are not ready or data are not available. Finally, users cannot misreport a late departure in a single-value domain because 0 is held if they obtain the results after the true d_i.

The interesting set is defined by:

$$I \subset \times_t \pi_i^t, t \in \{a_i, \cdots, d_i\} s.t. \sum_t \pi_i^t = l_i \tag{6}$$

Eq. (6) means that any allocation that includes l_i time units in the lifetime can satisfy the needs of user i. A shorter misreport $\hat{l}_i < l_i$ is impossible because inadequate allocated units result in a 0 value to the user. A longer misreport or adding garbage task $\hat{l}_i > l_i$ is also impossible. The allocation is not improved when all other parameters are fixed with $\hat{l}_i > l_i$ because of:

$$\hat{g}_i^t < g_i^t \tag{7}$$

where $\hat{g}_i^t = \frac{v_i}{\hat{l}_i - \lambda e_i^t}$, which means that

$$\hat{\pi}_i = 1 \Rightarrow \pi_i = 1 \tag{8}$$

when $\hat{l}_i > l_i$ and other parameters are fixed.
According to the payment rule, we know that

$$v_i^c = min \quad v_i \quad s.t. \quad \pi_i = 1 \tag{9}$$

We also know that

$$\hat{l}_i > l_i \Rightarrow \hat{v}_i^c > v_i^c \quad if \quad \hat{\pi}_i = 1, \pi_i = 1 \tag{10}$$

The users' misreport $\hat{l}_i > l_i$ result in worse allocation or higher payment. Hence, truthful report l_i is the dominant strategy.

If no VMs are allocated to User 1 in $[a_2, a_1]$ and $[d_1, d_2]$ when $\theta_1 \prec \theta_2$, then allocating l_1 units to User 1 requires that l_2 units be allocated to User 2 because of $g_1^t \leq g_2^t$ and $l_1 = l_2$ in $[a_1, d_1]$. If at least one VM is allocated to User 2 in $[a_2, a_1]$ or $[d_1, d_2]$ when $\theta_1 \prec \theta_2$, then allocating l_1 units to User 1 requires allocating l_2 units to User 2 because of $g_1^t \leq g_2^t$ and $l_1 > l_2'$ in $[a_1, d_1]$. Thus, our allocation rule is monotonic.

In summary, our mechanism M is truthful. □

4.4. Allocation and Payment Algorithm

The allocation and payment algorithms are described in Tables 4 and 5, respectively. $g_j^t(C+1)$ expresses the priority value of user j. This priority value is located in the $C+1$ position at time unit t.

4.5. Competitive Ratio

Because of the random arrival of users, the allocation strategy at t may not be the optimal choice in global opinion. Given period $[a, d]$, the ratio of the value sum between the offline optimal allocation and online allocation is worthy of concern. This ratio is defined as follows.

$$\begin{aligned} cr = max\frac{\sum_i v_i, \forall i, \pi_i^* = 1}{\sum_i v_i, \forall i, \pi_i = 1} \\ s.t. \forall t, \sum_i \pi_i^t \leq C, \sum_i \pi_i^{t^*} \leq C \end{aligned} \tag{11}$$

Table 4: Allocation algorithm

1.allocation()
2. $while(1)$
3. compute g_i^t for all effective user i
4. sort g_i^t in descending order
5. allocation one VM to forgoing C users respectively
6. t++
7. end while
8.end allocation

where π_i^* and $\pi_i^{t^*}$ refer to the offline optimum allocation, π_i and π_i^t refer to the online allocation, and *cr* is the competitive ratio, which expresses the worst case of offline optimum allocation against the online allocation in all situations. Offline optimum allocation is defined as:

$$\begin{aligned} &\pi^* = (\pi_1^*, \cdots, \pi_N^*) = arg \quad max \sum_i v_i \quad \forall i, \pi_i^* = 1 \\ &s.t. \forall t, \sum_i \pi_i^{t*} \leq C \end{aligned} \tag{12}$$

where all user types θ_i are known at the start time a in $[a, d]$. If the (12) is satisfied, the total user value is the maximum.
The following equation

$$\begin{aligned} cr' &= \frac{\sum_i v_i, \forall i, \pi_i^* = 1}{\sum_i v_i, \forall i, \pi_i = 1} \\ s.t. \forall t, &\sum_i \pi_i^{t*} \leq C, \sum_i \pi_i^t \leq C \end{aligned} \tag{13}$$

expresses the actual ratio in a deterministic case. The online allocation is generally not the worst case in this situation.

Theorem 2: Given $[a, d]$ satisfying $d - a = T - 1$, $C = 1$, $\lambda = 0$ and $[a_i, d_i] \subseteq [a, d]$, the competitive ratio will be $T - \varepsilon$.

Table 5: Payment algorithm

```
1.payment()
2.   while(1)
3.     if t==d_i
4.       compute v_i^c = g_j^t(C+1) * (l_i - λe_i)
           for all t, π_i^t == 1 and t ∈ [a_i, d_i]
5.       select the biggest v_i^c
6.       allocate repeatedly by v_i = v_i^c in [a_i, d_i]
7.       if π_i == 1 repeat 4.~6.
8.       end if
9.       record v_i^c
10.    end if
11.  end while
12.end payment
```

Proof. ∃ i, the user has the highest priority value $\frac{v_i}{l_i}$. Therefore, the worst case is that only a single task is completed and has the lowest value, which occurs with two users $\theta_1 = (a, d, T, v)$ and $\theta_2 = (t, t, 1, \frac{v}{T} + \varepsilon)$, where $t \in [a, d]$. The online allocation is $\pi_1 = 0$ and $\pi_2 = 1$, and the offline optimum allocation is $\pi_1 = 1$ and $\pi_2 = 0$, with a result of:

$$cr = \frac{v}{\frac{v}{T} + \varepsilon} = T - \varepsilon \tag{14}$$

□

Corollary 1: Given $[a, d]$ satisfying $d - a = T - 1$, $C = N$, $\lambda = 0$ and $[a_i, d_i] \subseteq [a, d]$, the competitive ratio will be $T - \varepsilon$.

The mechanism should be evaluated for a long period. Therefore, that $cr = T - \varepsilon$ is poor. By contrast, the mechanism satisfies $l_i << T$, so the worst case can never occur. Hence, we focus on the actual ratio cr' through a simulation that uses real records from data centers.

5. Performance Evaluation

To the best of our knowledge, publicly available cloud computing workloads are not currently available. Our simulation data were obtained from [24]. These included many workload logs from large scale parallel systems from various countries throughout the world. We used the RICC-2010 log, which derives from the RIKEN integrated cluster of clusters(RICC) that has operated since August, 2009. RIKEN is an independent scientific research and technology institution of the Japanese government.

More than 400 thousand records are included in the log, acquired over the course of several months. Parameters for this log include submitted time t_s, waiting time, running time t_r, number of allocated processors n_p and so on. All time is measured in s. The non-parallel form is modeled in our study. Therefore, a task with t_r running time using n_p processors is divided into n_p independent tasks each with t_r running time. One allocation time unit expresses 10 min in our simulation. For example, a record consists of $t_s = 1000$, $t_r = 5800$, and $n_p = 64$, which expresses 64 independent tasks, each task having $a_i = 3$, $l_i = 10$ in our simulation. Where $t_s = 1000$, it falls within the second unit, so the task participates in unit allocation from the third unit, expressed as $a_i = 3$. Where $t_r = 5800$, this means that at least 10 units are required to complete the task at least. Recalling the user's type $\theta_i = (a_i, d_i, l_i, v_i)$, d_i and v_i can not be acquired from the log directly but instead are generated through exponential distribution, where $d_i = a_i + l_i + l_i * exp(2)$ and $v_i = l_i * exp(50)$.

We selected 10,000 continuous records from the log. The duration of allocation and payment algorithms is from t_1^a to $max(t_{101}^d, \cdots, t_{10100}^d)$, where t_j^a and t_j^d denote the arrival time and deadline time units, respectively, of the tasks derived from the j^{th} record. All tasks that arrive during the aforementioned period participate in the algorithms. However, we record only the task valuations and payments whose arrival time unit is in $[t_{101}^a, t_{10100}^a]$. Some basic statistics related to the 10,000 records from the log are given in Table 6.

Table 6: Statistics of the 10,000 records

Total tasks	Average l_i	Average capacity
318265	30.3	4018

The average capacity is calculated by the following.

$$C_{avg} = \frac{\sum_i l_i}{t^a_{10100} - t^a_{101} + 1} \tag{15}$$

The arrival task distribution at each unit and the task distribution corresponding to l_i are shown in Fig. 1 and 2, respectively.

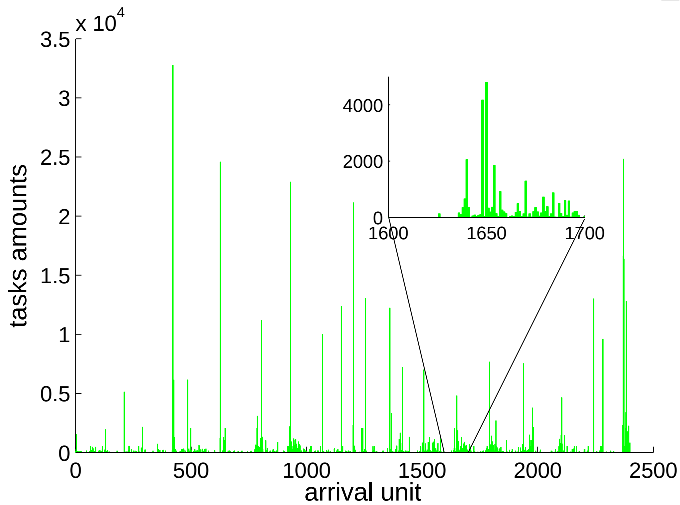

Figure 1: Task distribution at each unit

The offline optimal allocation in our problem is NP hard. An approximate method is adopted in our simulation. All task value densities (value per unit) that arrive during the algorithmic run are sorted by descending order. We select tasks set L satisfying

$$arg \quad max \sum_{i \in L} l_i \leq C * T \tag{16}$$

where C is the number of VMs and T is the duration units of the algorithm. The total value of the offline optimum allocation is calculated by summing the users' value arriving in $[t^a_{101}, t^a_{10100}]$ and contained in L. The total values acquired in offline optimal allocation and online allocation based on different VMs capacity are shown in Fig. 3. The total completed tasks according to different VMs capacity are shown in Fig. 4. Obviously, the total value and completed tasks all increase according to the increase in resource capacity. If

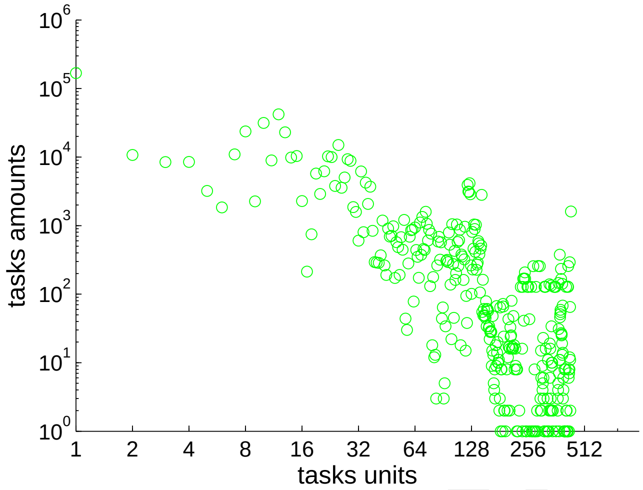

Figure 2: Task distribution to task-length units

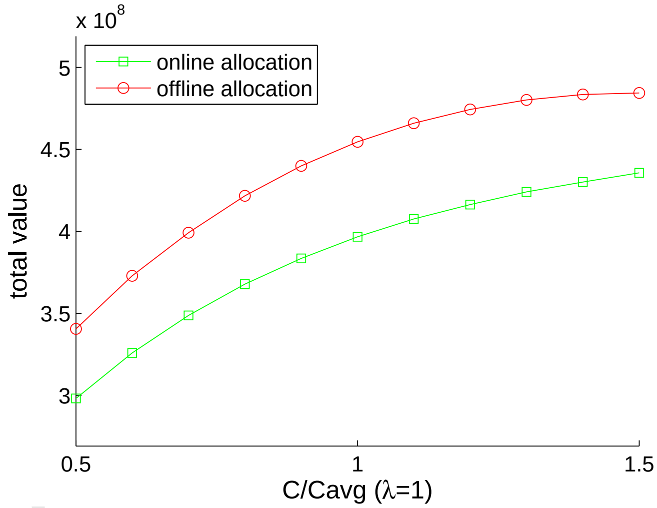

Figure 3: Total values according to different capacity

the provider's resource cost is known, a best resource capacity can be determined. The relationship between cr' and λ in a different resource capacity is illustrated in Fig. 5. We discover that the influence is obvious according to inadequate resource capacity. Although completing each unit's allocation and payment calculation in one unit time (10 minutes in our simulation) is easy, thousands of units exist and different parameters must be repeated

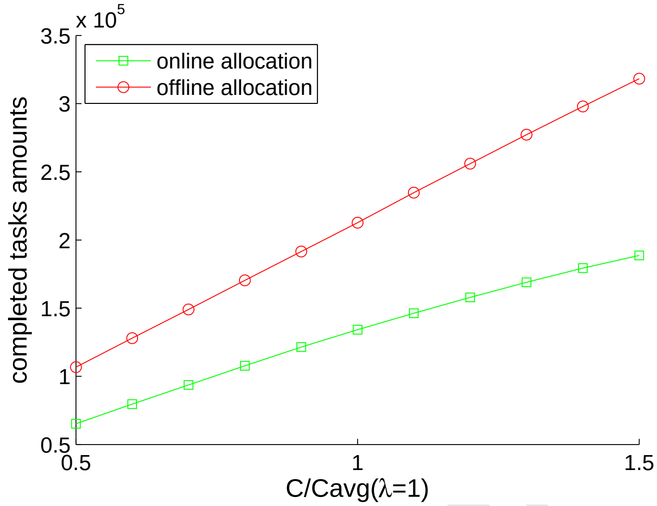

Figure 4: Total completed tasks according to different capacity

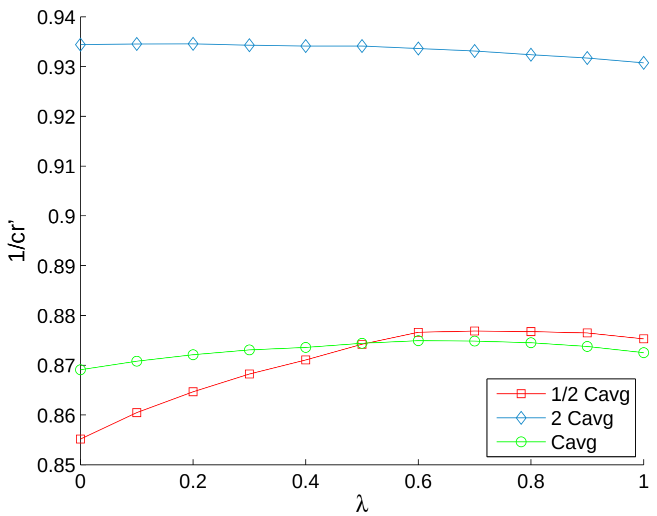

Figure 5: Actual competition ratio to λ in different capacity

several times. Therefore, we reduce the simulation requirement for an appropriate experiments time for the total payment. One thousand records are used, each of the 16 processor parallel tasks is equivalent to one non-parallel task, and resource capacity is also divided by 16. The total payment and total value according to different capacity are shown in Fig. 6.

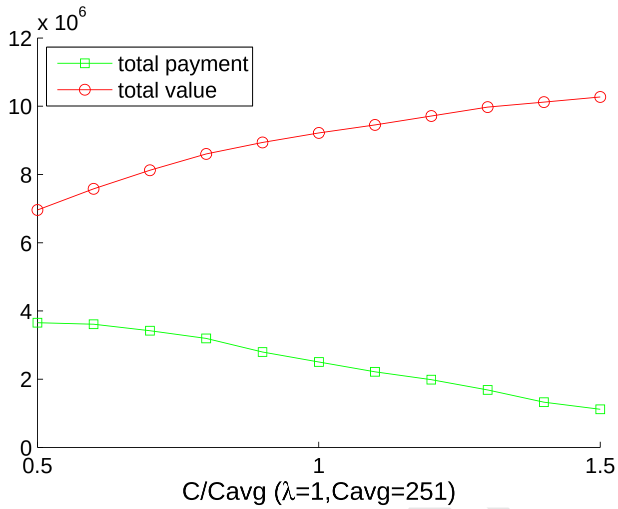

Figure 6: Total payment according to different capacity

The critical value is influenced by the number of users. Therefore, a random grouped pattern is proposed to examine the effect on payments. The cloud VMs pool is divided into several groups and tasks are randomly allocated to certain groups. We still apply our proposed mechanism M to each group. Obviously, user strategy misreports cannot turn to their interesting groups, so the mechanism remains truthful. A five-group simulation is run and a comparison is given in Fig. 7. Results show that the payments (cloud provider revenue) noticeably increase.

In summary, the total value and completed task amounts are positively correlated with the cloud resource capacity. The normally increasing speed of the total value slows down after $C/C_{avg} > 1$, and the private cloud provider can select an appropriate investment of the data center for economic benefits. A good λ can increase the efficiency of resources when the capacity is fixed. $\lambda = 0.6$ represents a quality choice when cloud resources are lacking. If private clouds are also interested in revenue, private cloud providers should construct a datacenter having limited capacity because the total payments decrease as resources capacity increases. The grouped preemptive online private cloud allocation mechanism represents an effective balance between total value and total payments.

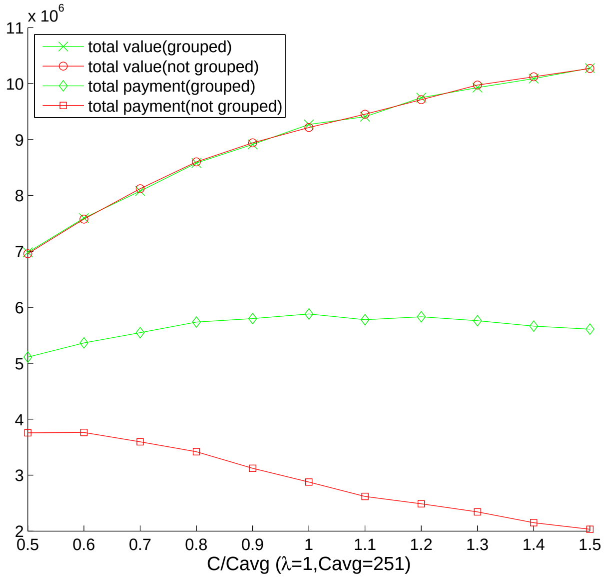

Figure 7: Comparison of general and grouped mechanism

6. Conclusions

HPC on cloud often employs pay-for-use as a resource strategy. Thus, an auction-based VMs allocation online mechanism is a promising solution for HPC cloud resource allocation-related problems, which are based on the randomly arriving and self-interested users. Promoting social welfare in private cloud is a reasonable goal, but user private information must be revealed. Therefore, in this study we proposed a novel truthful preemptive online mechanism for VMs allocation. We compared our mechanism with the offline optimal mechanism through social welfare using real tasks from a data center. Our mechanism is proven to have a superior actual competitive ratio. We evaluated the major parameter λ in the allocation rule. We also examined allocation efficiency between average workloads with the cloud capacity. Our mechanism is a useful solution for VMs allocation in private clouds.

Acknowledgments

The workload log from the RICC cluster was graciously provided by Motoyoshi Kurokawa. This work is supported by the National Natural Science

Foundation of China under Grant No. 61170029, 61472240, and 61373032, and the Zhejiang Provincial Science and Technology Plan of China under Grant No. 2013C31097.

[1] http://aws.amazon.com/ec2/spot-instances, accessed September 2014

[2] Yang,S.,Hajek,B.:"VCG-kelly mechanisms for allocation of divisible goods: adapting VCG mechanisms to one-dimensional signals",IEEE Journal on Selected Areas in Communications,2007,25,(6) pp. 1237-1243

[3] Jain,R.,Walrand,J.:"An efficient Nash-implementation mechanism for network resource allocation", Automatica,2010,46,(8) pp. 1276-1283

[4] Tang,W.,Jain,R.:"Hierarchical auction mechanisms for network resource allocation",IEEE Journal on Selected Areas in Communications,2012,30,(11) pp. 2117-2125

[5] Iosifidis,G.,Koutsopoulos,I.:"Double auction mechanisms for resource allocation in autonomous networks", IEEE Journal on Selected Areas in Communications,2010,28,(1) pp. 95-102

[6] Kong,Z.,Xu,C.,Guo M.:"Mechanism design for stochastic virtual resource allocation in non-cooperative cloud systems",Proc. 2011 IEEE International Conference on Cloud Computing (CLOUD),Washington DC,USA, July 2011,pp.614-621

[7] Wang L., Tao J., Ranjan R., Marten H., Streit A., Chen J., Chen D.:" G-Hadoop: MapReduce across distributed data centers for data-intensive computing", Future Generation Comp. Syst. 2013, 29(3), pp.739-750

[8] Wang L., Khan S.U., Chen D., Kolodziej J., Ranjan R., Xu C., Zomaya A.:" Energy-aware parallel task scheduling in a cluster", Future Generation Comp. Syst. 2013, 29(7), pp.1661-1670

[9] Wang L., Geng H., Liu P., Lu K., Kolodziej J., Ranjan R., Zomaya A.:" Particle Swarm Optimization based dictionary learning for remote sensing big data", Knowl.-Based Syst. 2015, 79, 43-50

[10] Song W., Wang L., Ranjan R., Kolodziej J., Chen D.:" Towards Modelling Large-Scale Data Flows in a Multidatacenter Computing System With Petri Net", IEEE Systems Journal, 2015, 9(2),416-426

[11] Prasad A.S., Rao,S.:"A mechanism design approach to resource procurement in cloud computing",IEEE Transactions on Computers,2014,63,(1) pp. 17-30

[12] Nejad,M.M.,Mashayekhy,L.,Grosu,D.:"A family of truthful greedy mechanisms for dynamic virtual machine provisioning and allocation in clouds",Proc. 2013 IEEE Sixth International Conference on Cloud Computing(CLOUD),Santa Clara Marriott,CA,USA,June 2013,pp. 188-195

[13] Zaman,S.,Grosu,D.:"Combinatorial auction-based allocation of virtual machine instances in clouds", Journal of Parallel and Distributed Computing,2013,73,(4) pp. 495-508

[14] Zhang,Q.,Zhu,Q.,Boutaba,R.:"Dynamic resource allocation for spot markets in cloud computing environments", Proc. 4th IEEE International Conference on Utility and Cloud Computing,Melbourne,Australia,December 2011,pp. 178-185

[15] Mashayekhy,L.,Nejad,M.M.,Grosu,D.:"A truthful approximation mechanism for autonomic virtual machine provisioning and allocation in clouds", Proc. 2013 ACM Cloud and Autonomic Computing Conference,Miami,Florida,USA,August 2013,no.9

[16] Zhang, L.,Li,Z.,Wu,C.:"Dynamic resource provisioning in cloud computing: A randomized auction approach",Proc. of IEEE INFOCOM 2014,Toronto,Canada,April-May 2014.

[17] Gerding,E.H.,Robu,V.,Stein,S.:"Online Mechanism Design for Electric Vehicle Charging",Proc. 10th International Conference on Autonomous Agents and Multiagent Systems,Taipei,Taiwan,May 2011,Vol.2,pp. 811-818

[18] Robu,V.,Stein,S.,Gerding,E.H.,et al.:"An online mechanism for multi-speed electric vehicle charging", Auctions, Market Mechanisms and Their Applications. Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering: Second International ICST Conference AMMA,New York,USA,August 2011,pp. 100-112

[19] Stein,S.,Gerding,E.,Robu,V.,et al.:"A model-based online mechanism with pre-commitment and its application to electric vehicle charging",Proc. of 11th International Conference on Autonomous Agents and Multiagent Systems,Valencia, Spain,June 2012,pp. 669-676

[20] Nisan,N.,Roughgarden,T.,Tardos,E.,et al.:"Algorithmic Game Theory",Cambridge University Press,New York, USA,2007

[21] Shi,W.,Zhang,L.,Wu,C.,et al.:"An online auction framework for dynamic resource provisioning in cloud computing", Acm Sigmetrics Performance Evaluation Review, 2014, 42(1):1.

[22] Zhang,H.,Li,B.,Jiang,H.,et al.:"A framework for truthful online auctions in cloud computing with heterogeneous user demands",Proc. of IEEE INFOCOM 2013,Turin,Italy,April 2013,pp.1510-1518

[23] Zaman,S.:"Combinat orial Auction-Based V irtual Machine Provisioning And Allocation In Clouds".PhD thesis,Wayne University,2013

[24] http://www.cs.huji.ac.il/labs/parallel/workload/logs.html,accessed June 2014