An Anti-cheating Bidding Approach for Resource Allocation in Cloud Computing Environments

Haiyang HU, Zhongjin LI, Hua HU

1School of Computer Science, Hangzhou Dianzi University, Hangzhou, China
2State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China

Abstract

Cloud computing is a large-scale distributed computing paradigm, which uses a usage-based pricing model to deliver various services and resources on demand to participants globally distributed over the Internet. Cloud computing systems incorporate different types of participants with varied private valuations for resources and multi-priority tasks which should be considered while designing effective resource allocation schemes for better share of resources. In addition, some participants may even be malicious with the goal to do damage to the system and increase their own utility. In this paper, we propose a bidding-based approach for resource allocation to address these issues. We investigate participants’ optimal bidding strategies in terms of the probability distributions that the valuations of their multi-priority tasks follow, and derive the symmetric Nash equilibrium solutions. In designing an efficient and anti-cheating resource allocation method for resisting the damage to the cloud computing system brought by malicious participants, we study the type of behavior of misreporting bidding function and give one statistical mechanisms to detect them. We also propose algorithms for helping participants to bid for resources in the environments and share their own resources. Finally, we conduct experiments to show the effectiveness of the proposed methods.

Keywords: Cloud Computing; Game Theory; Resource Allocation; Cheat Resisting

1 Introduction

In recent years, cloud computing has emerged as a new computing paradigm in distributed applications, mainly due to the advances in network, file sharing system, and distributed systems technologies [1-2]. Cloud computing systems have increased dramatically in their size, use, and underlying platforms, and they have been continuously focused on by researchers and engineers, such as IBM Blue cloud, Sun Cloud, Google Cloud and etc [3-5]. A fundamental purpose for these cloud computing systems is to provide various resources on demand to participants globally.
distributed over the Internet by applying a usage-based pricing mechanism. These resources include computation, data, storage, and their layered services. Participants can also provide their services as cloud computing resources to the environments [1] and [2].

To implement effective and secure resource allocation in cloud computing systems, several challenges must be addressed. First, as cloud computing systems are open and dynamic, they incorporate different types of participants with their private valuations for the public resources and multi-priority tasks, which should be considered while designing effective resource allocation schemes for better share of resources. For instance, a participant may hold several tasks with different priorities. For the high-priority tasks, e.g., some time-critical tasks, the participant urgently needs to obtain some resources to finish it immediately. While for the low-priority tasks, the participant may accept certain degree of delays in finishing these tasks, especially when resources are in a short supply or there are lots of other ones competing for the same resources. In addition, participants could be “selfish” for maximizing their own utility. Moreover, some participants may even be malicious and aim at doing damage to the system and other participants. Therefore, some special mechanisms are needed to detect these malicious participants and resist their harmful behavior.

Until recently, there have been many mechanisms proposed for resource allocation in different types of computing paradigms [6,11-20]. They can be categorized into three groups [7]: market-based, reciprocity-based and reputation-based approaches. In the market-based approaches [6], [11-13], and [17], participants submit bids for resources. Then, the resources are allocated due to some predefined resource allocation mechanisms. In the reciprocity-based approaches [18-21], each participant records and evaluates the behaviors of others that it interacts with, and then makes decision on how to serve another one by the direct service exchanges it receives from this participant. In the methods by using reputation-based approaches [14], [15], [16] and [20], the reputation of each participant is derived from its behavioral histories of providing services and consuming services. A participant having a good behavioral history of serving other ones is provided a service with better quality, when it competes with other users for some resources. Meanwhile, game theoretic methods have also been used to mathematically analyze cooperation in P2P networks, such as [22], [24-29]. In these methods, the authors attempted to make peers maintain a state in achieving a Nash equilibrium under which no one can improve their utility any more by deviating from its current strategy. However, we note that the reciprocity-based and reputation-based approaches may not be very applicable to cloud computing systems, which need the participators to pay for the resources they used.

In this paper, we propose to achieve the goal of effective and secure resource allocation in cloud computing environments by designing a bidding-based method under the game theoretic framework. We start by studying a simple scenario under which two participants bid for one resource, and then investigate the optimal bidding strategy. Next, we extend the analysis to a more general scenario under which multiple participants bid for a group of resources simultaneously, and derive the Nash equilibrium solutions. In designing an anti-cheating resource allocation mechanism for participants, we study the type of behavior of misreporting bidding function and give one statistical mechanisms to detect the malicious participants from the non-cooperative and noisy environments. Finally, we discuss the experiments conducted to show the effectiveness of the proposed methods in cloud computing environments.

The rest of the paper is organized as follows: Section 2 introduces our resource allocation (RA) game model among participants and the major notations used in this paper. Section 3 explores
the participants’ cheating strategies and presents several statistics-based mechanisms for detecting their malicious behavior. The algorithm is given in Section 4 to help participants’ make their bidding decisions and provide their own resources to the public, respectively. Section 5 conducts extensive experiments to evaluate the effectiveness of the proposed methods. Finally, Section 6 briefly concludes the work presented in this paper.

2 Our Game Model

In this paper, each resource, such as participant’s CPU cycles and the action to transmit a data packet for others, is treated as a discrete good and can be provided to environments by participants through auctions. When a participant needs resources to finish its tasks, it submits bids for these resources and competes with other buyers. Each participant autonomously generates its own tasks with different levels of priorities. Thus, the participants’ valuation of the same resource may be quite different according to their prioritized tasks. For example, if a participant holds a high-priority task, it will be inclined to pay more bids for the resource to finish the task. On the contrary, if the participant needs to finish a low-priority task, it may submit lower bids than its competitors do. In the following of this section, we will discuss the resource allocation game (RAG) for participants’ prioritized tasks in details. We use the first-price sealed-bids auction [8] in the game, i.e., when several participants compete for one resource, the participant who submits the highest bid becomes the winner and obtains this resource.

In our resources allocation game, each task $a_i$ generated by a participant $pa_i$ has a priority denoted as $\text{priority}(a_i)$. We also use $v_i$ to denote $pa_i$’s monetary valuation for $a_i$ in terms of its priority($a_i$), which is typically application-specific, and remains known by its competitors. Suppose now a resource $r$ is provided to a group of participants, and it can be used to finish their tasks. Each $pa_i$ will have a different valuation of $r$ according to the evaluation of its own task $a_i$. Without losing generality, we use $v_i$ to denote $pa_i$’s monetary valuation of $r$ for finishing its task $a_i$. In this paper, we focus on achieving a symmetric Nash equilibrium, where the bidding function $\beta$ is the same for all the participants. The bid submitted by $pa_i$ for $r$ is defined by $b_i = \beta(v_i)$. Finally, no participants can increase their payoff by taking other bidding strategies deviating from the current optimal bidding strategy in achieving the symmetric Nash equilibrium.

First, we introduce our resource allocation game (RAG). In such a game, participants’ valuations are at least $\bar{v}$ with $\bar{v} \geq 0$, and at most $\overline{v}$, which is a common knowledge known by all participants. Thus, $[\underline{v}, \overline{v}]$ is the interval over which each participant’s valuation $v_i$ ranges. If $N$ represents the number of participants in a resource allocation game, and $S_i = \{s_i|s_i \in [\underline{v}, \overline{v}]\}$ be the bidding strategy space of participant $pa_i$, then the bidding strategy combination of all participants can be denoted by $s = (s_1, \cdots, s_N)$, where $S = \times_{1 \leq i \leq N}S_i$ is the participants’ strategy space. Furthermore we use $s_{-i}$ to denote the strategy combination of all peers except $pa_i$:

$$s_{-i} = (s_1, \cdots, s_{i-1}, s_{i+1}, \cdots, s_N)$$

Next, we use $U_i(s)$ to denote the utility of $pa_i$ under the strategy combination $s$. Finally, a strategy combination, $s^*$, is said to achieve the state of Nash equilibrium, if and only if the following holds:

$$U_i(s^*) \geq U_i(s^*_{-i}, s_i) \forall s_i \in S_i, 1 \leq i \leq N$$

(1)
Note that, if a strategy combination is said to achieve the state of *Nash equilibrium*, then no peer can improve its utility by unilaterally deviating from its own strategy.

### 2.1 Two-participant resource allocation game

We first study a simple yet illustrative two-participant RAG, i.e., two participants compete for one resource. In this game, there are two participants (*pa*1 and *pa*2) competing for one resource *r* provided by a resource provider *pa*3 in the environment. Suppose in the current stage that each *pa*i (*i* = {1, 2}) has generated one task *a*i which can be finished by using *r*. In the auction for *r*, each *pa*i independently submits its bid *b*i for *r* based on its valuation *v*i. During the auction, each participant has no knowledge about the other one’s valuation of *r*. Nevertheless, some information needs to be provided to the participants for helping them make the bidding strategy.

Let *F*i(*x*) be the probability distribution function that *pa*’s valuation for resources follows on [v, v̄], which can be obtained from its data logs over a long period of time. Without losing generality, it is assumed that *F*i(*x*) is continuous on [v, v̄], and is defined by

\[
F_i(V) = \Pr(v_i \leq V) = \int_{v}^{V} f_i(x) \, dx
\]

where \(V \leq v\) and *f*i(*x*) is the corresponding probability density function (pdf) of *F*i(*x*) on [v, v̄].

In the auction for *r*, *pa*1 and *pa*2 are required to report their *F*i(*v*i) to *pa*3. Then, *pa*3 broadcasts the information of \{*F*i(*x*)\} to the two participants to help them develop the bidding strategies. In addition, we note that some participant may misreport its *F*i(*x*). A statistics-based mechanism is given in Section 3.1 for checking whether the participants honestly report their *F*i(*v*i). Furthermore, we also conduct experiments on this mechanism for demonstrating its effectiveness.

Now, we analyze the *Nash equilibrium* in this two-participant RAG. We assume that if *pa*i obtains *r* by paying a bid *b*i, its payoff is \(e_i = v_i - b_i\).

For a rational participant, it explicitly holds that \(v_i > b_i\) (Otherwise, the participant can not gain any payoff in finishing its task). Thus, the payoff gained by *pa*i in the auction of *r*, can be defined as: 1) \(e_i = v_i - b_i\), if *b*i > *v*i, i.e., *pa*i obtains *r* by the auction; 2) \(e_i = (v_i - b_i)/2\), if *b*i = *v*i, i.e., if *pa*i happen to submit a bid which is the same as the other one does, *pa*3 then the winner will be randomly selected between the two participants; 3) \(e_i = 0\), if *b*i < *v*i, i.e., *pa*i submits less bid for *r*, and then loses in the auction. Therefore, the expected payoff gained by *pa*i is:

\[
e_i = (v_i - b_i) \cdot \Pr(b_i > b_j) + \frac{1}{2} \cdot (v_i - b_i) \cdot \Pr(b_i = b_j)
\]  

(2)

In this two-participant RAG, let the bidding profile \(b = (b_1, b_2)\) be the strategy combination of *pa*1 and *pa*2, and \(b^* = (b_1^*, b_2^*)\) be the strategy in achieving a symmetric *Nash equilibrium* in this game. The value of \(b_i^*(i = 1, 2)\) can be given by the following theorem:

**Theorem 1:** In two-participant RAG with private valuations, if each *pa*’s valuation *v*i (*i* = 1, 2) independently follows a probability distribution *F*i on [v, v̄], then the bidding strategy \(b^* = (b_1^*, b_2^*)\) in achieving the state of a symmetric *Nash equilibrium* can be given by \(b_1^* = v_1 - \frac{\int_{v_2}^{v_1} F_2(x) \, dx}{F_2(v_1)}\) and \(b_2^* = v_2 - \frac{\int_{v_1}^{v_2} F_2(x) \, dx}{F_2(v_2)}\).
Proof: To achieve the symmetric Nash equilibrium, let \( \beta(v_i) \) be the bidding function taken by participant \( pa_i \). Suppose that \( pa_i \)'s valuation of the resource \( r \) is \( v_i \), and it submits a bid \( b_1 = \beta(v_i) \) for \( r \). Then, the expected payoff gained by \( pa_1 \) is \((v_1 - b_1)Pr(b_2 < b_1) + \frac{1}{2}(v_1 - b_1)Pr(b_1 = b_2)\), which leads to \((v_1 - b_1)Pr(\beta(v_2) < b_1) + \frac{1}{2}(v_1 - b_1)Pr(b_1 = b_2)\). Let \( \beta^{-1} \) be the inverse function of bidding function \( \beta \). Then the expected payoff gained by \( pa_1 \) can be rewritten by \((v_1 - b_1)Pr(v_2 < \beta^{-1}(b_1)) + \frac{1}{2}(v_1 - b_1)Pr(b_1 = b_2)\).

The probability of participant \( pa_2 \)'s bid being at most \( \beta^{-1}(b_1) \) is \( F_2(\beta^{-1}(b_1)) \). Thus, the expected payoff obtained by \( pa_1 \) is \((v_1 - b_1)F_2(\beta^{-1}(b_1)) + \frac{1}{2}(v_1 - b_1)Pr(b_1 = b_2)\), and its optimal bidding strategy is

\[
b_1^* = \arg \max_{b_1} ((v_1 - b_1)F_2(\beta^{-1}(b_1)) + \frac{1}{2}(v_1 - b_1)Pr(b_1 = b_2)) \tag{3}
\]

Therefore, the best bidding strategy taken by participant \( pa_1 \) is to choose \( b_1^* \) which can maximize its payoff in (3). Thus, the derivative of (3) on \( b_1 \) should be zero: \(-F_2(\beta^{-1}(b_1)) + (v_1 - b_1)\frac{F_2'(v_1)}{\beta'(v_1)} = 0\), which leads to

\[
\beta'(v_1)F_2(v_1) + \beta(v_1)F'_2(v_1) = v_1F'_2(v_1)
\]

By integrating both sides of the equation, we obtain

\[
\beta_1^* = \beta(v_1) = v_1 - \int_{v_1}^{b_1^*} \frac{F_2(x) dx}{F_2(v_1)}
\]

In symmetry, for participant \( pa_2 \), we can also obtain its optimal bidding strategy, \( b_2^* = v_2 - \int_{v_2}^{b_2^*} \frac{F_1(x) dx}{F_1(v_2)} \).

Note that the result presented in theorem 1 is different from the result presented in [8], in which all the bidders should have the same probability distribution of \( F_i(x) \), i.e., it should have \( F_1(x) = F_2(x) \).

### 2.2 N-participant resource allocation game

Now, we study a more general scenario where there are \( n (n > 2) \) participants bidding for a public resource \( r \), which we call a \( N \) - participant resource allocation game. Assume that in stage \( t \), each \( pa_i \) holds a set of tasks \( A_i(t) = \{a_i^{m}\} \) and each \( a_i^{m} \) has an valuation \( v_i^{m} \). Let \( c_i(t) = |A_i(t)| \) be the number of these tasks. Then, for each task \( a_i^{m} \in A_i(t) \), \( pa_i \) needs to submit a bid \( b_i^{m} \) in order to get the necessary resources to finish it. Thus, the total amount of payoff gained by \( pa_i \) with bidding for resources to finish its tasks in stage \( t \), can be given by

\[
E_i(t) = \sum_{m=1}^{c_i(t)} (v_i^{m} - b_i^{m}) p_i^{m} \tag{4}
\]

where \( p_i^{m} \) is used to denote whether \( pa_i \) can successfully obtain a resource to finish its task \( a_i^{m} \). More specifically, if \( pa_i \) can do this, \( p_i^{m} = 1 \); otherwise, \( p_i^{m} = 0 \).

On the other hand, each \( pa_i \) can also play a role of resource provider, i.e., in stage \( t \), \( pa_i \) may have a group of available resources \( R_i(t) = \{r_i^{k}\} \), and it can sell them out by auction to other participants. Let \( g_i^{k} \) denote the payoff gained by \( pa_i \) through selling out a resource \( r_i^{k} \) in stage \( t \),
and $d_i(t) = |R_i(t)|$ be the number resources. Then, $G_i(t)$, the total amount of payoff gained by $pa_i$ through selling its resources in stage $t$, is given by

$$G_i(t) = \sum_{k=1}^{d_i(t)} q_i^k q_i^k$$

where $q_i^k$ is used to denote whether $pa_i$ successfully sells out its resource $r_i^k$, i.e., if $q_i^k = 1$, then $r_i^k$ is sold out; otherwise, $q_i^k = 0$.

Next, the total amount of utility gained by $pa_i$ in stage $t$ can be given by:

$$U_i(t) = E_i(t) + G_i(t)$$

Proof: Suppose that there are $n$ participants bidding for a public resource $r$ simultaneously, and the expected payoff gained by each $pa_i$ is

$$e_i = (v_i - b_i) \prod_{j=1, j \neq i}^{n} \Pr(\beta(v_j) < b_i)$$

which leads to $(v_i - b_i) \prod_{j=1, j \neq i}^{n} F_j(\beta^{-1}(b_i))$. Note that, in equation (7), we omit the payoff gained by $pa_i$ under such a scenario that some other participants pay the same highest bids as $pa_i$ does, and $pa_i$ is lucky enough to be selected as the winner. However, as the probability of $\Pr(b_i = b_j)(i \neq j)$ is close to zero and omitting such a payoff will not influence the result at all when we compute its derivative, we do not include this case in Eq. (7)

For any $pa_i$, its optimal bidding strategy $b_i^*$ is

$$b_i^* = \arg\max_{b_i} (v_i - b_i) \prod_{j=1, j \neq i}^{n} F_j(\beta^{-1}(b_i))$$

That is, the derivative of (8) on $b_i$ should be zero:

$$- \prod_{j=1, j \neq i}^{n} F_j(\beta^{-1}(b_i)) + (v_i - b_i) \sum_{k=1, k \neq i}^{n} F_j(\beta^{-1}(b_i)) \frac{F'_k(v_i)}{\beta'(v_i)} = 0$$

which leads to the following equation:

$$\beta'(v_i) \prod_{j=1, j \neq i}^{n} F_j(v_i) + \beta(v_i) \sum_{k=1, k \neq i}^{n} \prod_{j=1, j \neq i, k}^{n} F_j(v_i) F'_k(v_i) = v_i \sum_{k=1, k \neq i}^{n} \prod_{j=1, j \neq i, k}^{n} F_j(v_i) F'_k(v_i)$$

By integrating both sides of the equation, we obtain

$$b_i^* = \beta(v_i) = v_i - \frac{\int_{v_i}^{v_u} \prod_{j=1, j \neq i}^{n} F_j(x) dx}{\prod_{j=1, j \neq i}^{n} F_j(v_i)}$$
Note that the result presented in theorem 2 is different from the result presented in [8], in which all the bidders should have the same probability distribution of \( F_i(x) \), i.e., it should have \( F_i(x) = F_2(x) = \cdots = F_n(x) \).

**Theorem 3:** \( \beta(v_i) \) is a increasing function on \([\underline{v}, \overline{v}]\).

**Proof:** it can be easily proved from (9).

**Theorem 4:** When a participant \( \text{pa}_i \) takes its optimal bidding strategy in achieving the Nash equilibrium, its expected payoff monotonously grows with the increase of its valuation \( v_i \) on \([\underline{v}, \overline{v}]\).

**Proof:** From (9), the expected payoff gained by \( \text{pa}_i \) is

\[
(v_i - \beta(v_i)) \prod_{j=1, j \neq i}^n F_j(\beta^{-1}(b_i)) = \frac{\int_{\underline{v}}^{\overline{v}} \prod_{j=1, j \neq i}^n F_j(x) \, dx}{\prod_{j=1, j \neq i}^n F_j(v_i)} \prod_{j=1, j \neq i}^n F_j(v_i) = \int_{\underline{v}}^{v_i} \prod_{j=1, j \neq i}^n F_j(x) \, dx
\]

Since each \( F_j(x) \) is a cumulative function and increases monotonously on \([\underline{v}, \overline{v}]\), we immediately conclude that the expected payoff gained by \( \text{pa}_i \) increases with the increase of its valuation \( v_i \) on \([\underline{v}, \overline{v}]\).

Next, we analyze the existence of NE in a \( n \) – participant RAG.

**Theorem 5:** In the \( n \) – participant RAG, there exists a Nash equilibrium.

**Proof:** Let \( e_i \) be the payoff that \( \text{pa}_i \) can gain in a \( n \) – participant RAG. \( v_i \) and \( b_i \) are its valuation and bid in the game, respectively. From (7) and (8), the second-order derivative of \( e_i \) on \( b_i \), can be given by

\[
e_i'' = - \sum_{k=1, k \neq i}^n \prod_{j=1, j \neq i, k} F_j(v_i) F_k^\prime(v_i)
\]  

(10)

In (10), since each \( F_i \) is the cumulative distribution function, it has \( F_i(x) > 0 \) and \( F_i'(x) = f_i(x) > 0 \) on \([\underline{v}, \overline{v}]\). Then, it has \( e_i'' < 0 \), i.e., \( e_i \) is concave with \( b_i \). Next, the domain of \( \text{pa}_i \)'s bidding strategy is the interval \([\beta(\underline{v}), \beta(\overline{v})]\), which is explicitly a compact convex set. Therefore, by Rosen’s theorem [10], there exists an NE in the \( n \) – participant RAG.

Now, we present another important result as follows:

**Theorem 6:** In \( n \)-participant RAG with private valuations, each \( \text{pa}_i \)'s optimal bid \( b_i^* \) rises in price with the increasing number of its competitors.

**Proof:** we start from (9) to prove it \( \frac{\int_{\underline{v}}^{\overline{v}} \prod_{j=1, j \neq i}^n F_j(x) \, dx}{\prod_{j=1, j \neq i}^n F_j(v_i)} \). Let , where \( v_i \geq \underline{v} \). Then \( b_i^* \) can be rewritten by

\[
b_i^* = v_i - e(n, v_i)
\]  

(11)

Now, we study \( e(n, v_i) - e(n + 1, v_i) \), which is given by

\[
e(n, v_i) - e(n + 1, v_i) = \frac{\int_{\underline{v}}^{\overline{v}} \prod_{j=1, j \neq i}^n F_j(x) \, dx}{\prod_{j=1, j \neq i}^n F_j(v_i)} - \frac{\int_{\underline{v}}^{\overline{v}} \prod_{j=1, j \neq i}^{n+1} F_j(x) \, dx}{\prod_{j=1, j \neq i}^{n+1} F_j(v_i)} = \frac{1}{\prod_{j=1, j \neq i}^{n+1} F_j(v_i)} \int_{\underline{v}}^{\overline{v}} \prod_{j=1, j \neq i}^n F_j(x)(F_{n+1}(v_i) - F_{n+1}(x)) \, dx
\]  

(12)
For any \(pa_i\), its \(F_i(x)\) is a cumulative function of its valuations and monotonously increases on \([v_i, \overline{v}]\), i.e., for any \(x \in [v_i, v_i]\), it holds that \(F_{n+1}(v_i) \geq F_{n+1}(x)\). Therefore, it holds that \(\int_{v_i}^{v_i} \prod_{j=1, j \neq i}^{n} F_j(x)(F_{n+1}(v_i) - F_{n+1}(x)) > 0\).

Thus, we obtain \(e(n, v_i) - e(n+1, v_i) > 0\), i.e., \(e(n, v_i)\) monotonously decreases with \(n\) increases. Next, from (11), we obtain the result that the value of \(b^*_i\) rises with the increasing number of its competitors.

Now, we present a result about a participant’s payoff and its valuation \(v_i\) in the \(n\)-participant RAG.

**Theorem 7:** For any \(pa_i\), the ratio of its payoff \(e(v_i)\) to its valuation \(v_i\) grows with \(v_i\) increasing.

**Proof:** The payoff obtained by \(pa_i\) in an auction can be given by \(e(v_i) = v_i - b^*_i = \int_{v_i}^{v_i} \prod_{j=1, j \neq i}^{n} F_j(x) \, dx\). Let \(\phi(v_i)\) be the ratio of \(e(v_i)\) to \(v_i\). It can be calculated below. \(\phi(v_i) = \frac{e(v_i) \int_{v_i}^{v_i} \prod_{j=1, j \neq i}^{n} F_j(x) \, dx}{v_i}\). Then, we obtain:

\[
\phi'(v_i) = \frac{v_i \prod_{j=1, j \neq i}^{n} F_j(v_i) - \int_{v_i}^{v_i} \prod_{j=1, j \neq i}^{n} F_j(x) \, dx}{(v_i)^2} \tag{13}
\]

Let \(h(v_i) = v_i \prod_{j=1, j \neq i}^{n} F_j(v_i)\), and the numerator of equation (13) can be further represented by \(\mu(v_i) = h(v_i) v_i - \int_{v_i}^{v_i} h(x) \, dx\). And the derivative of \(\mu(v_i)\) on \(v_i\) is given by

\[
\mu'(v_i) = h'(v_i) v_i \tag{14}
\]

It evidently holds that \(\mu'(v_i) > 0\), and it indicates that \(\mu(v_i)\) monotonously increases on \([v_i, \overline{v}]\). When \(v_i = v\), we obtain \(\mu(v) = h(v) v > 0\). Therefore, we have \(\phi'(v_i) = \frac{\mu(v_i)}{(v_i)^2} > 0\). Then, \(\phi(v_i)\) also monotonously increases on \([v_i, \overline{v}]\).

### 3 Malicious-behavior Detection Mechanisms

Because cloud computing environments are dynamic and open, some participants may be malicious with the goal to do damage to the system. In this section, we focus on the scenario where the system containing a finite number of participants will keep running for an enough long time. Each participant will stay in the system for a reasonably long time. We also assume that each participant has a unique registered identity, which can be verified by a central authority (CA). This identification will be used to perform necessary access control and authentication. CA also manages the information of all the auctions occurring in the system, i.e., after a successful auction, the resource provider reports the details of the auction process to CA, which includes the identities of the bidders participating in the auction, their bids and the final winner. Thus, CA can use this information to detect the malicious participants. In addition, there exists a central bank (CB) in the system, which stores each participant’s virtual currency, deposit and transaction. It also fairly exchanges virtual currency between resource providers and bidders.

For a malicious \(pa_i\), its goal is to increase its own utility by degrading the system performance and wasting the resources of other participants as much as possible. In general, \(pa_i\) can achieve its goal by taking such a harmful action, \(pa_i\) misreports its \(F_i(x)\). In an RAG, a rational \(pa_i\) can derive its optimal bidding strategy \(b^*_i\) according to Eq.(9). So, it needs to know in advance
the \( \{F_i(x)\} \) of other participants who also join in the auction. Thus, in our RAG, each \( pa_i \) must honestly report its \( F_i(x) \) to the resource provider. However, some malicious participants may misreport their \( F_i(x) \) to disturb other participants’ bidding strategies and do damage to their utility. We need some approaches to detect this kind of cheating behavior.

In an auction, each rational participant can derive its optimal bidding function from the \( \{F_i(x)\} \) of other participants according to Equation (9). So, the system needs each participant to honestly report its \( F_i(x) \) to the resource provider and CA. However, since some malicious participants may misreport their \( F_i(x) \), we need to devise a method to detect their cheating behaviors. Based on chi-square goodness-of-fit test [9, 31], a mechanism called \( DM_F \) is discussed in this section to detect this type of cheating behavior. The idea of \( DM_F \) is to check whether \( pa_i \)'s actual bidding behavior is consistent with its \( F_i(x) \) reported, i.e., bids submitted by \( pa_i \) in the RAGs should be rational regarding its \( F_i(x) \) according to (9).

In \( DM_F \), let \( M \) represent the total number of resource allocation games \( \{RAG_1, \cdots, RAG_M\} \) which \( pa_i \) had participated. Let \( b_i^m \) represent the bid submitted by \( pa_i \) in an auction \( RAG_m (1 \leq m \leq M) \), which was also reported by the resource provider to CA. Next, for this \( b_i^m \), CA assumes that \( pa_i \)'s bidding strategy in \( RAG_m \) should be optimal, and it derives \( pa_i \)'s valuation \( v_i^m = \beta^{-1}(b_i^m) \) according to (9). Then, each \( v_i^m \) is viewed as a sample \( Y_m \) in \( DM_F \), and CA now has a group of such samples \( \{Y_1, \cdots, Y_M\} \) with a size \( M \). Finally, CA only needs to test whether the samples \( \{Y_1, \cdots, Y_M\} \) follow \( F_i \) on \([\underline{y}, \bar{y}]\).

To do this, CA starts by postulating two hypotheses:

- Null hypothesis \( H_0 : \{Y_1, \cdots, Y_M\} \) follows \( F_i(x) \), which indicates that \( pa_i \) honestly reports its \( F_i(x) \);

- Alternative hypothesis \( H_1 : \{Y_1, \cdots, Y_M\} \) does not follow \( F_i(x) \), which indicates that \( pa_i \) misreports its \( F_i(x) \).

CA then divides the interval \([\underline{y}, \bar{y}]\) which \( \{Y_1, \cdots, Y_m\} \) distributes on into \( D \) disjoint intervals, \( I_1, \cdots, I_D \). Let \( O_j \) be the actual number of samples located in \( I_j \), and \( EN_j \) be the expected number of samples locating in \( I_j \), i.e., \( EN_j = M \int_{I_j} f_i(x) \, dx \) (Here, \( f_i(x) \) is the corresponding pdf of \( F_i(x) \)). Then, CA obtains

\[
\chi^2 = \sum_{j=1}^{D} \frac{(O_j - EN_j)^2}{EN_j} \tag{15}
\]

If \( M \) is large enough (i.e., \( M \geq 50 \)), Eq.(15) will approximately follow \( \chi^2(D - 1) \) [9, 31]. Let \( \alpha \) be the significance level, that is,

\[
\Pr(H_0 \text{ is true but } H_0 \text{ is resused}) = \alpha
\]

For \( \chi^2 \) in (15), if it has \( \chi^2 \geq \chi^2_\alpha(D - 1) \), CA immediately refuses the hypothesis \( H_0 \), and \( pa_i \) is identified to have misreported its \( F_i(x) \).

In general, \( DM_F \) can be used to detect whether a participant’s bidding behavior is consistent with its \( F_i(x) \).
4 Resource Allocation among Participants

In our algorithm, there are three phases in which participants bid for the resources they need: In the first phase, each participant seeks those non-malicious resource providers who currently provide the resources it needs, and then registers with the providers and plan to bid for their resources. In the second phase, the providers announce the information about its bidders to the public, which includes the number of bidders who have participated in its RAG, as well as their \{F_i(v_i)\}. Then, if a bidder finds another resource which can be used to finish its task and increase its payoff, it will adjust its bidding strategy by choosing to bid for this resource. The second phase continues until the deadline is reached or the total payoff gained by participants can not be increased anymore.

To analyze this process quantitatively, we use an indicator \( x(a^m_i, r^m_j) \) to denote whether \( pa_i \) can successfully obtain resource \( r^m_j \) and finish its task \( a^m_i \), that is, if \( pa_i \) obtains \( r^m_j \), then \( x(a^m_i, r^m_j) = 1 \); otherwise, \( x(a^m_i, r^m_j) = 0 \). Let \( e(a^m_i, r^m_j) \) be the corresponding payoff gained by \( pa_i \) in finishing \( a^m_i \). Then, the total amount of payoff, \( E_i(t) \), gained by \( pa_i \) in \( t \) is given by

\[
E_i(t) = \sum_{j=1}^{N} \sum_{m=1}^{c_i(t)} \sum_{n=1}^{d_j(t)} x(a^m_i, r^m_j) e(a^m_i, r^m_j) \quad (16)
\]

Let \( E^K_i(t) \) denote the total payoff gained by \( pa_i \) in the \( k-th \) adjustment on its bidding strategy. Then, for some \( \varepsilon > 0 \) set by \( pa_i \) itself, if it observes that

\[
|E^K_i(t) - E^{K+1}_i(t)| < \varepsilon \quad (17)
\]

The \( pa_i \) will not change its bidding strategy any more.

The algorithm for a participant to make its bidding strategies in time stage \( t \) is given in Figure 1.
5 Experiments

In this section, we present experimental analysis on our resource allocation approaches. We test the effects of different parameters on the bidding results. The simulations are developed with C++ programming language and Matlab 7.0. In the experiments, the number of participants is set to be \( N = 30 \). Each participant’s valuation, \( F_i \), follows the uniform probability distribution on \([0, 1]\), i.e., \( \underline{v} = 0 \) and \( \overline{v} = 1 \). \( c_i(t) \), the number of tasks generated by one participant in each stage is randomly selected in . In the experiments, we studied on the participants’ payoff and their successfully bidding rate (SBR) (i.e., the rate at which a participant \( p\alpha_i \) can successfully obtain the resources for finishing its tasks).

5.1 Valuation of participants

In this section, we conduct a set of experiments to study the values of participants’ valuations. In the experiments, the number of tasks generated by each participant in one stage is randomly selected in the range of . The system has run for 1000 time stages. We change the participants’ valuation \( \{v_i\} \) in three different ranges, i.e., \( v_i \in [0, 1] \), \( v_i \in [0.5, 1] \) and \( v_i \in [0.5, 0.6] \), and then studied the participants’ payoff and their SBR.

Some results are shown in figure 2 and 3. With the value of \( v_i \) increasing, the payoff gained by each participant also grows. More specifically, in figure 2, when the value of \( v_i \) varies in , participants can gain the maximum payoff. This result conforms to Theorem 4 as proposed earlier. Next, we identified that the payoff gained by participants given \( v_i \in [0, 1] \) is larger than those gained by participants given \( v_i \in [0.5, 0.6] \). The reason is that when \( v_i \in [0, 1] \), participants bidding for the same resource may have very different valuations distributed on \([0, 1]\). Then, those participants that have higher valuations will pay at higher bids for the resources, and win the resources in the end. Thus, according to Theorem 4, these participants can gain more payoffs in the auction. In addition, we also observe that the value of \( v_i \) has little impact on participants’ SBR.
5.2 Detecting participants’ malicious behaviors

In this section, we conduct experiments on the statistical mechanisms proposed in Section 3, which are used for checking whether participants misreporting its $F_i(x)$. In the experiments, we compared the average payoff and $SBR$ gained by each non-malicious participant under two different malicious scenarios, that is, the scenario where the system uses $DM_F$ proposed in Section 3 to detect the malicious behaviors (shown in figure 4(c) and (d)), and the scenario where the system does not use $DM_F$ (shown in figure 4(a) and (b)). The percent of malicious participants, $\omega$, is set in such a way that it increases from 10% to 80%. By using $DM_F$, the parameter $\alpha$ is set to increase from 0.02 to 0.08, and the number of time stages is fixed at 1000. The $F_i(x)$ of each $pa_i$ is a uniform probability distribution on. However, each malicious $pa_i$ misreports its $F_i(x)$ as a uniform probability distribution on a different domain $[0.8,1]$.

In the experiments, we record the ratio of non-malicious participants’ payoff and $SBR$ when there are $\omega$ percent of malicious ones in the system to their payoff and $SBR$ when there are no malicious ones in the system, and they are represented by $\eta_{\omega}(\text{payoff})$ and $\eta_{\omega}(SBR)$, respectively. As shown in figure 4 (a) and (b), when $\omega$ increases, the values of $\eta_{\omega}(SBR)$ and $\eta_{\omega}(\text{payoff})$ are reduced greatly. This is because the non-malicious participants bidding at rational prices have little change to obtain the resources in the auctions with their valuations distributed on . However, when the system uses $DM_F$ to detect the cheating behaviors, the malicious participants will be very careful to submit their bids to avoid being detected. In this way, the non-malicious participant can gain more payoff and $SBR$. For example, even if the $\omega$ increases up to 0.8, the
non-malicious participants can obtain more than 96 percent of the payoff that they may obtain in a non-malicious scenario.

6 Conclusion

In this paper, we presented a market-based approach for allocating resources among the autonomous participants with multi-priority tasks. We also proposed a detecting approach for identifying malicious participants in a non-cooperative and noisy cloud computing environments. We first study at a simple scenario under which two participants bid for one public resource, and investigated the Nash equilibrium in terms of the probability distributions that participants’ valuations for their tasks follow. Then, we extended the analysis to a more realistic scenario under which multiple participants bid for a group of resources, and derive the Nash equilibrium solutions. In designing an efficient and security resource allocation algorithm for resisting the malicious participants, we study a malicious behavior that participants could have, which is misreporting $F_i(x)$. A statistical mechanism is also presented for detecting the malicious participants from the noisy environment. Finally, we conducted a set of experiments to illustrate the effectiveness of the proposed mechanisms.

References


