

Research Article

A Game-Theoretic Based Resource Allocation Strategy for Cloud Computing Services

Wang Yan,¹ Wang Jinkuan,¹ and Sun Jinghao²

¹College of Information Science and Engineering, Northeastern University, Shenyang, Liaoning 110819, China

²College of Computer and Communication Engineering, Northeastern University at Qinhuangdao, Qinhuangdao, Hebei 066004, China

Correspondence should be addressed to Wang Yan; wangyan@mail.neuq.edu.cn

Received 9 May 2016; Revised 14 August 2016; Accepted 23 August 2016

Academic Editor: Jun Zheng

Copyright © 2016 Wang Yan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose an economics-oriented cloud computing resources allocation strategy with the use of game theory. Then we develop a resource allocation algorithm named NCGRAA (noncooperative game resource allocation algorithm) to search the Nash equilibrium solution that makes the utility of various resource providers achieve optimum. We also propose an algorithm named BGRRA (bargaining game resource allocation algorithm) to further increase the overall revenue with the constraints of efficiency and fairness. Based on numerical results, we discuss the influence of NCGRAA and BGRRA for the utility of resource on the system performance. It shows that the choice of parameters of the two algorithms is significant in improving the system performance and converging to the Nash equilibrium and Nash bargaining.

1. Introduction

Cloud service is the commercial implementation of computing resources and essentially a producers-consumers model. From the perspective of economics, the service is as the valuable economic goods that providers produce and lease to consumers. Cloud users can purchase goods from the provider according to their own needs in global scope and in a certain paying mode.

Advancement in cloud resource allocation is urgently needed to satisfy service-specific QoS (Quality of Service) requirements. At the same time, system performance must be mathematically analyzed and numerically evaluated for design of the strategies. The generally accepted view is that typical cloud computing resource allocation strategies are mainly based on multistage resource pool of rent-borrow theory, the random integer programming optimal resource strategy, and economics resource strategy.

Our research on cloud computing resources provisioning strategy is based on the market economic model and the advantages of game theory in resource management. First, we establish a noncooperative resource provision model and a

corresponding game algorithm. The aim of the algorithm is to provide resources strategy by solving equilibrium so that each utility function could achieve optimum to obtain a reasonable strategy. Meanwhile, in order to further increase the overall revenue, the strategy is proposed by taking the efficiency and fairness constraints in noncooperative bargaining game into account. The according algorithm achieves higher collective utility by solving bargaining problem. Finally, experiment verifies the effectiveness and superiority of the two algorithms.

In Section 2, we analyze the related work. In Section 3, we analyze the framework of cloud resources based on game theory and establish the market model of resources provided. In Section 4, we propose an algorithm named NCGRAA to solve the Nash equilibrium in order to obtain reasonable resource allocation strategy. In Section 5, we propose an algorithm named BGRRA to search the Nash bargaining solution in order to make higher utility of resources. In Section 6, based on simulation, we evaluate the performance of the two algorithms. Finally, concluding remarks are given in Section 7.

2. Related Work

Researchers [1, 2] used a random integer programming method to optimize resource allocation and introduced two ways which are in advance order and pay-as-you-go to provide users with resources. This strategy can provide dynamic resource allocation methods, considering the cost of various stages resources. However the uncertain resource prediction accuracy needs to be improved.

Then the cloud resource management model [2, 3] was designed based on SLA (Service Layer Agreement). Although the perspective of economics has been taken into account, this strategy is still lacking factors like the resources and the price dynamic. Literature [4] had studied the resource management system based on QoS.

In recent years, various resource management methods with different policies and principles were proposed in some literatures [5–14]. But most literatures were relevant to complex cloud based on applications consisting of multiple subtasks. Therefore collaboration between tasks should be considered. Meanwhile, existing algorithms were obviously too conservative without considering the communications among tasks. As we know, the dependent task scheduling problem is NP-hard [15–17]. Thus some heuristic algorithms were proposed for near-optimal solution [18, 19]. Simultaneously, numerous methods were proposed to solve the optimization problem with game theory [20, 21]. Moreover, game theory can be divided into noncooperative game theory and cooperation game theory [22]. Typical noncooperative game solution is the Nash equilibrium.

Based on the Nash equilibrium, numerous works were proposed. For instance, Kwok et al. [23] proposed a level game model under the environment of multiple administrative domains. It studied the selfish behavior of resource for the influence of user task execution. However, the research does not put the relationship between the resource providers into consideration. In literature [24], a distributed actively non-cooperative abandon model was presented. The model reduced the task execution time considering the selfish and priority of resource allocation. Ghosh et al. [25] proposed a pricing strategy solving the Nash bargaining problem based on game theory. But the uniqueness of Nash equilibrium solution was not discussed.

Existing works generally consider the relationship between the user and resource provider, but the resource provider is neglected as rational body when the relationship between competition and cooperation on the result of final resources is provided. Researchers proposed resource management on the basis of the principle of market economics [26] and focused on how to improve the benefits of resource provider with the constraints of efficiency and fairness.

To the best of our knowledge, although the resource allocation strategies have been studied in previous works, allocation strategy aims at achieving the optimal allocation of resources that has been less proposed. Therefore, we focus on giving the dynamic change among supply-demand, resource load, and congestion by prices floating reflections and also focus on achieving the optimal resources allocation

by equilibrium theory. Thus we propose the supply strategy for cloud computing resources based on market economics model in this paper and two resources allocation algorithms for increasing both efficiency and fairness. Inspecting our experimental results, the proposed algorithms improve the resource allocation performance effectively. The experimental results indicate that the resources allocation problems that are solved by noncooperative theory could converge to the Nash equilibrium solution. The utility of the resource provider is optimal. While bargaining, noncooperative game analysis with Pareto improvements is made on the basis of satisfying the efficiency and fairness. Moreover, two algorithms are effective to improve the efficiency of resource allocation.

3. The Game Theory for Resource Allocation Strategy

As utility computing, cloud computing has a lot of economy features, including market features of computing resource, the scale of the cloud market economy, and the provider paid features of resources. One of the advantages of using economics principle to solve the cloud resource allocation is the market mechanism. The market mechanism is well known as that commodity prices floating reflections resource load, congestion, and the dynamic change of supply or demand condition. Therefore it can achieve the optimal allocation of resources with equilibrium theory. This dynamic coordination of resources allocation provides a mechanism that is suitable for dynamic characteristic of the cloud computing environment. Each participant in the market is in pursuit of maximizing individual interests. That makes the whole resource allocation gradually tend to optimum condition. It is one of the traditional resources allocation methods pursued but difficult to achieve.

Hence we analyze the situation above and give the architecture based on economics. The cloud architecture is shown in Figure 1. It has four parts: the users, resource consumers, resource market, and resource providers.

(1) *Users*. We mean the users who use cloud computing resources. Cloud users put forward their own requirements from their agent to the cloud market, including the required QoS description. And this requires many aspects to complete the task, such as CPU type, the number of the CPU, memory size, operating system, and its version number.

(2) *Resource Consumers*. Private cloud is as a resource consumer, even though it provides users with resources. When it cannot meet users' need, it still gets resources from the public cloud.

(3) *Resource Market*. Resource market is the core of this architecture module. It is the management for users, it calculates risk, and it is autonomous. The market is the interface for data center/cloud service providers and the user/agent, using some kind of negotiation mechanism to coordinate the two requirements. It mainly includes the following six kinds of mechanisms.

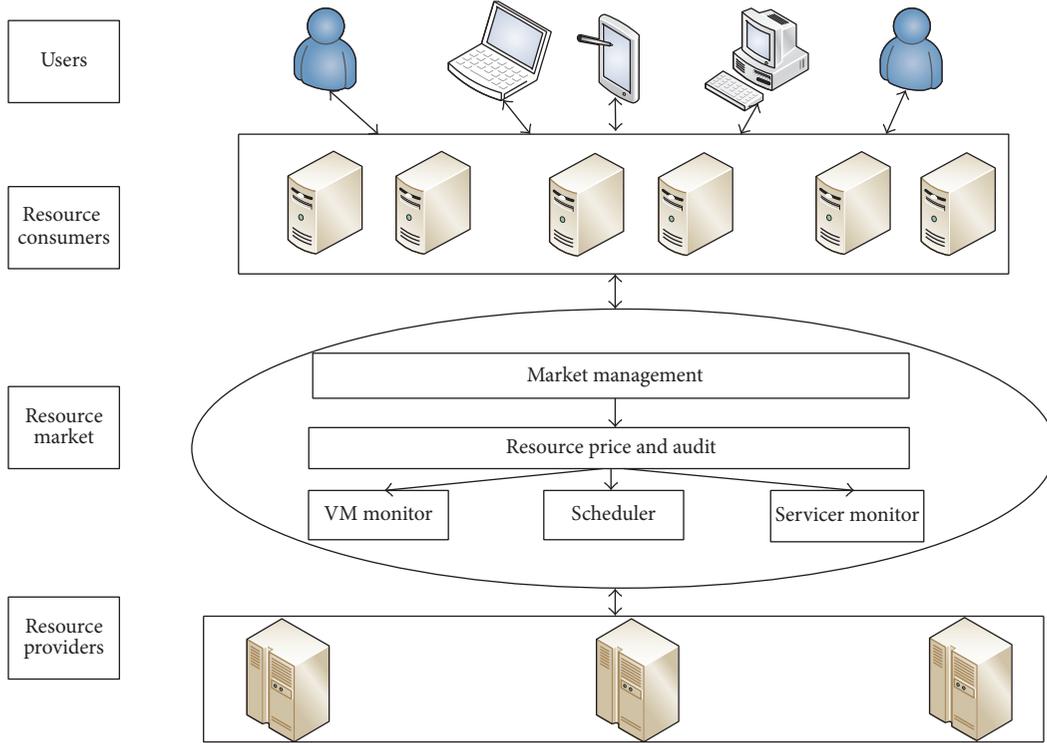


FIGURE 1: The cloud architecture based on economics.

Permission control mechanisms for service request: when a service request is submitted for the first time, the mechanism submits request for QoS explanation before determining whether to accept or reject the request. Pricing mechanism: pricing as a basic mechanism manages the resources allocation and demand of data center. Pricing mechanism decides the price of service request. Audit mechanism: audit mechanism records the actual use of the requested resources, in order to make the final cost that can be calculated and credited for the user. VM (Virtual Machine) monitoring mechanism: monitoring mechanism monitors VM available. Scheduler: it allots and executes the service requests. Monitoring system for service request: the service request monitoring mechanism monitors the implementation of the service request.

(4) *Resource Providers.* The public cloud is represented.

We build the system model with the $M/M/n/n+l$ queue. The model has some features: the model is a mixed queue model. Tasks arrive at the queue in accordance with the parameter λ of the Poisson flow and service time for each task obeys the negative exponential distribution with parameter μ . The order of the service is in accord with First Come First Served. Length of the queue is $n+l$ and l is the length of buffer. If task arrives at the system, when the queue has not been occupied, it enters the queue to wait for service or it waits in the buffer. Parameters of the model are shown in Parameters section.

(1) *Response Time.* It has proved that when the communication intensity $\rho < 1$, there is a stable equilibrium state in

system [12, 13]. Therefore, we consider the queue for $\rho < 1$. Probability π_k of queue length is k : $\pi_k = \lim_{t \rightarrow \infty} P\{N(t) = k\}$.

Typically, we measure response time with the percentage φ , which is in response to a predetermined percentage of the time. In the SLA (Service Layer Agreement), performance indicator of response time is described as follows:

$$\int_0^{\Delta T} f(t) dt \geq \varphi, \quad (1)$$

where the $f(x)$ is a random function and is also the probability density function. ΔT represents the period during which task uses the service.

Equation (1) represents the probability of response time; when the response time is less than the period ΔT , then the probability must be equal to or more than φ . For example, in $M/M/1$ queuing model, the probability density function of the response time is $f(t) = \mu(1 - \rho)e^{-\mu(1-\rho)t}$. Thus we can give the equation as follows:

$$\mu \geq \frac{-\ln(1 - \varphi)}{\Delta T} + \lambda. \quad (2)$$

From (2), we know that, when the arrival rate λ is constant, the time period ΔT is unchanged. If we improve the probability of response time φ , service rate will increase. We must deploy more computing resources.

In $M/M/n/n+l$ queue model, system transition probability between states is as follows:

$$\pi_k = \begin{cases} \frac{(n\rho)^k}{k!} \pi_0, & 0 \leq k < n \\ \frac{n^n \rho^k}{n!} \pi_0, & n \leq k \leq n+l. \end{cases} \quad (3)$$

When $\sum_{i=1}^{n+l} \pi_i = 1$, we obtain π_0 as follows:

$$\pi_0 = \left(\sum_{k=0}^{n-1} \frac{(n\rho)^k}{k!} + \frac{(n\rho)^n (1 - \rho^{l+1})}{n! (1 - \rho)} \right)^{-1}, \quad \rho < 1, \quad (4)$$

where π_0 means the probability that all hosts are idle and there is no task in system.

Response time is the sum of waiting time and service time. Waiting time can be calculated in the following manner method: When a task arrives, if the buffer queue is not full, the user can enter the queue to wait. q_i represents the steady probability that the number of tasks waiting for service is i . q_i is given as follows:

$$q_i = \frac{P\{N(t) = i\} P\{l \neq 0\}}{P\{l \neq 0\}} = \frac{\pi_i}{1 - \pi_{n+l}}, \quad (5)$$

$$i = 0, 1, \dots, n+l-1.$$

When the buffer queue is not full, hosts service the tasks immediately; then the probability distribution at this time is as follows:

$$W_a(t) = q_0 + q_1 + q_2 + \dots + q_{n-1} = \sum_{i=1}^{n-1} \frac{\pi_i}{1 - \pi_{n+l}}. \quad (6)$$

When a task arrives at the buffer, if there are tasks already in the queue, the newly arriving task has to wait until its front tasks have been completed. Distribution of exit tasks stream for each host is following negative exponential distribution with parameter μ . When the number of the hosts is n , the system distribution for exit tasks stream is subject to negative exponential distribution with parameter $n\mu$. In these circumstances, we derive the Erlang distribution as follows:

$$\begin{aligned} W_a(t) &= W_a(0) + \sum_{i=n}^{n+l-1} P\{0 < W_a \leq t\} \\ &= W_a(0) + \sum_{i=n}^{n+l-1} q_i \int_0^t \frac{n\mu (n\mu x)^{i-n}}{(i-n)!} e^{-n\mu x} dx \\ &= \sum_{i=1}^{n-1} \frac{\pi_i}{1 - \pi_{n+l}} + \sum_{i=n}^{n+l-1} q_i \int_0^t \frac{n\mu (n\mu x)^{i-n}}{(i-n)!} e^{-n\mu x} dx. \end{aligned} \quad (7)$$

Response time is equal to the waiting time and the service time; consider the following:

$$\begin{aligned} W(t) &= P\{W \leq t\} = \int_0^t P\{W_a \leq t-x\} \mu e^{-\mu t} dx \\ &= \int_0^t W_a(t) \mu e^{-\mu t} dx. \end{aligned} \quad (8)$$

By responding to the distribution function of time, the relationship between the percentage of the number of servers and QoS can be obtained. If the response time in the percentage of ΔT is φ , then $W \geq \varphi$; consider $T < 0.05$; it indicates that the response time is less than 0.05 seconds.

(2) *The Average Queue Length.* Depending on the desired definition, the average queue length in the buffer can be written as follows:

$$\begin{aligned} \bar{N}_a &= \sum_{k=n}^{n+l} (k-n) \pi_k \\ &= \frac{n^n \rho^{n+1} \pi_0}{n! (1-\rho)^2} (1 - (l+1)\rho^l + l\rho^{l+1}), \quad \rho < 1. \end{aligned} \quad (9)$$

The average queue length is equal to the queued requests waiting for service and the sum of the number of requests. The average queue length is as follows:

$$\begin{aligned} \bar{N} &= \bar{N}_a + \bar{N}_s = \bar{N}_a + \rho(1 - \pi_{n+l}) \\ &= \frac{n^n \rho^{n+1} \pi_0}{n! (1-\rho)^2} (1 - (l+1)\rho^l + l\rho^{l+1}) \\ &\quad + n\rho \left(1 - \frac{n^n \rho^{n+l} \pi_0}{n!} \right) \\ &= \frac{n^n \rho^{n+1} \pi_0 (1 - (l+1)\rho^l + l\rho^{l+1})}{n! (1-\rho)^2} \\ &\quad + n\rho \left(1 - \frac{n^n \rho^{n+l} \pi_0}{n!} \right). \end{aligned} \quad (10)$$

According to queuing model proposed above, when all hosts in private cloud have been fully allocated, VM manager buys VMs resources. For this situation, we give the system benefits function as follows:

$$F(c, h') = \delta N' - [\epsilon W_a + c(h' - n) + En], \quad (11)$$

where c represents the price of buying VMs in the market. h' represents the number of hosts being demanded. δ and ϵ represent constant factors. N' represents the throughput of processing tasks. E represents the cost of energy of provided VMs in a private cloud.

The goal of this paper is to solve the provisioning strategy to ensure the best interests of resource providers in different relations, based on the assumption of considering the market as a whole.

Resource consumer demand F_c is a function of the price c . Demand function is as follows:

$$F_c = \arg \max F(c, h'). \quad (12)$$

For the resource requirements of a private cloud, the demand function is as follows:

$$F_d(c) = \sum_{k=0}^I h'_{d,k} \chi_{\Xi_{d,k}}(c) = \begin{cases} \sum_{k=0}^I h'_{d,k}, & c \in \Xi_{d,k} \\ 0, & \text{otherwise,} \end{cases} \quad (13)$$

where I indicates the total number of steps. $h'_{d,k}$ represents the number of VMs that buyer needs at the stage of step k . Ξ is the price range at step k . $\Xi_{d,k} = (\Phi_{d,k-1}, \Phi_{d,k}]$, where $\Phi_{d,k}$ is the right margin. d is the number of buyers. For $d \in [1, 2, \dots, D]$, $\Xi_{d,0} = 0$, $\Xi_{d,1} = (0, \Phi_{d,1}]$. $\chi_{\Xi_{i,k}}$ is the index function of demand.

$$\begin{aligned} \chi_{\Xi}(c) &= \begin{cases} 1, & c \in \Xi \\ 0, & \text{otherwise} \end{cases} \\ \Phi_i &= \Phi_{\hat{d}, \hat{k}} \\ h' &= \sum_{d=1}^D F_{(c, h')}(\Phi_i) \\ \text{s.t. } (\hat{d}, \hat{k}) &= \arg \min_{\Phi_{i,k'} \in \mathbb{Q}_k} \Phi_{i,k'} \\ \mathbb{Q}_k &= \{(\Phi_{i,k'}) \mid \Phi_{i,k'} > \Phi_{k-1}\}. \end{aligned} \quad (14)$$

Therefore, the total consumer demand is

$$F(c) = \sum_{k=0}^I h'_k \chi_{\Xi_k}(c). \quad (15)$$

From the above analysis, we get the total demand price function. That is an inversion of the consumer demand function, given as follows:

$$F^{-1}(c') = \sum_{k=0}^I \Phi_{I-1} \chi_{\Xi_k}(c') = \begin{cases} \sum_{k=0}^I \Phi_{I-1}, & c' \in \Xi_k \\ 0, & \text{otherwise,} \end{cases} \quad (16)$$

where $\Phi_0 = 0$, $\Xi_k = (\Phi_{k-1}, \Phi_k]$, $\Xi_I = (\Phi_{I-1}, \infty]$, and $\chi_{\Xi}(c')$ is the index function of price:

$$\chi_{\Xi}(c') = \begin{cases} 1, & c' \in \Xi \\ 0, & \text{otherwise.} \end{cases} \quad (17)$$

4. Noncooperative Game Based NCGRAA

In the market, the situation that multiple cloud resource providers compete with each other may be noncooperative game model. Noncooperative game theory is perfectly rational in the context of the strategy with interdependence interaction theory. Particularly, noncooperative game theory focuses on solving the problem for rational stakeholders from multiple rows cooperative game resources, mutual influence, and effect arising from the analysis of the resources. The most excellent choice is a function that all other competitors selected. Nash equilibrium strategies seek to optimize the effectiveness of the time optimized configuration of resources.

We express resource providers providing VMs as space strategy. In cloud computing market, the behavior of all the resources providers can be represented as Cartesian product $S = S_1 \times S_2 \times \dots \times S_o$, where S_o represents a resource provider o 's strategy set.

Utility represents obtained satisfaction by providing a service or a resource rent and each resource provider's expected revenue is represented by the utility function, that is, $\{U_o\} = \{U_1, U_2, \dots, U_o\}$, $o \in \{1, 2, \dots, O\}$. s is the choice strategy of the resource providers in the game. If the game participant o is with the strategy s_o , $s_{o'}$ is the representation of selection strategy by all other resource providers except o and $s_{o'} = s - s_o$. Then the participant utility is expressed as $U_o(s)$.

In the cloud computing market, all the resource providers want to obtain the most optimal strategy according to price function. Considering the case of a single resource provider, the returns function is as follows:

$$u(s) = sF_{(c, h')}^{-1}(s) - F_c(s). \quad (18)$$

Among them, the right first term indicates the total income provided by the VMs. Introducing cost function can improve the system performance and it is defined as follows:

$$F_c(s) = sF_{uf} - F_f, \quad (19)$$

where F_{uf} is the variable costs of resource providers providing VMs and it includes the cost of the buffer. F_f is the fixed costs of resource providers providing VMs.

Strategic optimization can be formalized as

$$s^+ = \arg \max_s u(s). \quad (20)$$

According to (20), benefits of single resource provider are as follows:

$$u^+(s) = s^+ F_{(c, h')}^{-1}(s^+) - F_f - s^+ F_{uf}, \quad (21)$$

where price function is nonincreasing step function. The number of steps is finite and it can be solved by enumeration method.

For a cloud resource market with resource providers o , given other resource providers VMs, the strategy of VMs providing of resource provider o is optimal. For all resources providers, the following equation is given:

$$U_o(s_o^+, s_{o'}) \geq U_o(s_o, s_{o'}). \quad (22)$$

There is the following optimization problem solution:

$$s_o^+ \in \arg \max_{s_o \in S_o} U_o(s_o, s_{o'}). \quad (23)$$

Suppose resource providers provide alternatives with fixed costs and there is variable cost. Marginal cost is zero. The utility function is

$$U_o = sF_{(c, h')}^{-1}(s) - F_f^{(o)} - sF_{uf}^{(o)}, \quad (24)$$

where $F_f^{(o)}$ is fixed cost and $F_{uf}^{(o)}$ is variable cost.

In order to obtain Nash equilibrium strategy, we describe Nash equilibrium as follows: we find out the best solution for each player in game and then figure out strategy combination composed of the best strategy for each player.

Optimal response strategy means that the strategy is optimal among the given strategies of all other resource providers.

$$s_o^+ = \arg \max U_o(s_o, s_{o'}), \quad (25)$$

where $U_o(s_o, s_{o'}) = s_o F_{(c,h')}^{-1}(s_o + s_{o'}) - F_f^{(o)} - s_{o'} F_{uf}^{(o)}$.

As the price function is nonincreasing step function, we obtain that

$$\begin{aligned} s_o^+ &= h_{k^+}' - s_{o'} \\ j^+ &= \arg \max_{j \in \{1, 2, \dots, j'\}} [\Phi_k(h_k' - s_{o'}) - F_f^{(o)} - (h_k' - s_{o'}) F_{uf}^{(o)}] \\ \text{s.t. } j' &= \arg \min_{h_k' \in \mathbb{Z}} h_k' \\ Z &= \{(h_k') \mid h_k' > s_{o'}\}. \end{aligned} \quad (26)$$

Using the optimal response, Nash equilibrium is defined as

$$\begin{aligned} s_{o'}^+ &= f(\vec{s}_{o'}^+) \\ \vec{s}_{o'}^+ &= [s_1^+, s_2^+, \dots, s_{o-1}^+, s_{o+1}^+, \dots, s_O^+]^T, \end{aligned} \quad (27)$$

where $\vec{s}_{o'}^+$ represents the optimal response vector.

At this time, utility function with Nash equilibrium strategy of resource provider o represents the receipts of noncooperation market resulting as follows:

$$R_{MR}^+ = U_o(s_o^+, \vec{s}_{o'}^+). \quad (28)$$

Theorem 1. *If a task selects a VM which is not in the possible set of optional VMs, the task scheduling scheme is not a Nash equilibrium solution.*

Proof. The current task number is greater than or equal to the VMs set, so there is no task that can be chosen, except the set of VMs. The number of tasks is less than that of the VMs, assuming that a task i is scheduled to VM except the possible set. Then according to the possible set of VMs, besides i task, the number of rest tasks is assigned as $N - 1$. According to the principle of Nash, there must be at least one VM which may not be assigned, so, for this task, $U_i > U_{oth}$. U_i is the benefit for the reason that the task i is scheduled to the possible VM and U_{oth} is the benefit for the reason that the task i is scheduled to the VM not in the possible set.

Because Nash equilibrium is defined under the condition of invariable environment, the entire individual cannot achieve higher yields by changing its own strategy. The formula below is tenable:

$$\forall k, f(s_1, \dots, s_i, \dots, s_n) \geq f(s_1, \dots, s_k, \dots, s_n), \quad (29)$$

where s_i is the strategy for the task i and $f(s_1, \dots, s_i, \dots, s_n)$ is the benefit at the strategy s_i .

It is at odds with the definition of Nash equilibrium solutions, so if task i selects the possible VMs, the task scheduling scheme is not Nash equilibrium solution of the scheduling problem. \square

Theorem 2. *In the noncooperative game with a finite number of players model, for $o \in \{1, 2, \dots, O\}$, sufficient conditions for the existence of Nash equilibrium are the following:*

- (1) Strategy space $\{S_o\}$ is a nonempty compact convex set in Euclidean space.
- (2) Utility function $U_o(s)$ is continuous in $\{S_o\}$ and is quasiconcave function.

Proof. As the number of VMs available is limited for each resource provider in terms of a single integer point set, its strategy space for real axis is clearly a nonempty compact convex set in Euclidean space. Because the number of VMs is an integer number and greater than zero, the price function is as the number of linearly decreasing functions at this time. The utility function is continuous in space strategy map.

$$\begin{aligned} \frac{\partial U_o(s_o, \vec{s}_{o'})}{\partial s_o} &= F_{(c,h')}^{-1}(s_o) + s_o \frac{\partial F_{(c,h')}^{-1}(s_o)}{\partial s_o} \\ &\quad - \frac{\partial F(s)}{\partial s_o} \end{aligned} \quad (30)$$

$$\frac{\partial^2 U_o(s_o, \vec{s}_{o'})}{\partial s_o^2} = 2 \frac{\partial F_{(c,h')}^{-1}(s_o)}{\partial s_o} + \frac{\partial^2 F_{(c,h')}^{-1}(s_o)}{\partial s_o^2},$$

where the price function is a decreasing linear function and $\partial^2 U_o(s_o, \vec{s}_{o'}) / \partial s_o^2 < 0$. Then $U_o(s)$ is concave in the strategy space.

By Theorem 2, it shows that there is Nash equilibrium strategy for cloud resources provided in a competitive market.

$$\begin{aligned} \frac{\partial U_o(s_o, \vec{s}_{o'})}{\partial s_o} = 0 &\implies \\ s_o &= \frac{-F_{(c,h')}^{-1}(s_o) + \partial F(s) / \partial s_o}{\partial F_{(c,h')}^{-1}(s_o) / \partial s_o}. \end{aligned} \quad (31)$$

By the above formula, Nash equilibrium can be represented as a matrix form.

$$\mathbf{E}(\mathbf{s}) = \mathbf{C}\bar{\mathbf{s}} + \mathbf{Z}, \quad (32)$$

where $\mathbf{E}(\mathbf{s})$ represents a mapping from the space \mathbf{R}^O to \mathbf{R}^O . \mathbf{C} and \mathbf{Z} are zero-order matrixes.

Nash equilibrium is to prove that the uniqueness of $\mathbf{E}(\mathbf{s})$ is a standard function that satisfies the following:

- (1) Positive: $\mathbf{E}(\mathbf{s}) > 0$.
- (2) Monotonic: if $x < y$, then $\mathbf{E}(x) \geq \mathbf{E}(y)$.
- (3) Scalability: for any $\omega > 1$, there is $\omega \mathbf{E}(x) \geq \mathbf{E}(\omega x)$.

Since each cloud resource provider in the market obtains certain benefits through the VMs providing, namely, $\mathbf{E}(\mathbf{s}) > 0$, that meets (1). Proving the existence of the process, because the price is understood to be a linear decreasing function, we can see that the nature (2) is satisfied. For a linear function, a function of the nature (3) scalability is satisfied.

To sum up, $E(s)$ is a standard function and, for a standard function, fixed point is unique. Therefore, Nash equilibrium of NCGRAA algorithm is unique.

The process of algorithm of NCGRAA is as follows.

Step 1. When there is a set of a new request arriving, it makes this set of VMs have the maximum profit and minimum completion time of the VMs.

- (a) According to the system running state and the tasks users submit, compute the running time and make sure that each task has the profit of each VM.
- (b) Sort each task profit and find out each set for the task.
- (c) Summarize set.

Step 2. Find all possible task groups with the intersection and generate the corresponding conflict task group. Perform the noncooperative game analysis and then identify Nash equilibrium for all conflict task groups and generate for each conflict task group the Nash equilibrium solution. Generate Nash equilibrium solution set for all conflict task groups, the Nash equilibrium solution for task scheduling problem and optimal Nash equilibrium solution.

- (a) Analyze the task groups. After that, find out the tasks with intersection and generate the corresponding conflict task group.
- (b) For each conflict set, find out the Nash equilibrium solution and generate the Nash equilibrium solution set and the Nash equilibrium solution for the corresponding task scheduling problem.
- (c) According to the system revenue for Nash equilibrium solution to the current task scheduling problem and the overall completion time for task, find out the optimal Nash equilibrium solution for the current task scheduling problem.

Step 3. According to the task scheduling result with the optimal Nash equilibrium, each task is assigned to the selected VM. The allocation results are recorded in task scheduling log.

- (a) According to the task scheduling result with the optimal Nash equilibrium, determine the service VM and perform the task.
- (b) Record the task scheduling results in the log.

Step 4. Continue monitoring a user action, running the task scheduling algorithm for the next set of task scheduling requests. \square

5. Bargaining Game Based BGRAA

Although NCGRAA makes resource provider optimal profit, it is not the biggest gain. If resource providers can mutually agree, it will have more collective gains. The kind of bargain in the market resource provider of income will be greater than or equal to the cooperative competition in the market returns.

In a negotiated market, resource providers improve their income by the way of negotiating. Market can benefit from the

bargaining game model. The resource providers can mutually agree to bargain to get the satisfaction, which has the two aspects, both efficiency and fairness of the VM provided on the basis of strategy. Efficiency requirements can contain Pareto efficiency and the overall revenue maximization. The solution is defined as Nash bargaining equilibrium of bargaining game.

5.1. Nash Bargaining Game Model for Resource Allocation

Definition 3. All the resource providers in the cloud computing market environment are defined as the game participants, expressed as $RP_o, o \in \{1, 2, \dots, O\}$.

Definition 4. B is the set of VM bargaining problem, $F \in \mathbf{R}^O$ is the set space of resource provider, and Γ is the minimum expected value for earnings. $\Gamma = (R_{MR}^{(1)*}, \dots, R_{MR}^{(o)*}, \dots, R_{MR}^{(O)*})$.

Definition 5. When $\{R_{bar}^{(o)} \in F \mid R_{bar}^{(o)} \geq R_{MR}^{(o)*}, \forall o\}$ is not empty bounded set, $(F, \Gamma) \in \mathbf{B}$ is the bargaining game problem.

A bargaining solution is to put each bargaining problem $(F, \Gamma) \in \mathbf{B}$ linked with a special result $R_{bar}^* = \psi(F, \Gamma)$ as a function, named $\psi : \mathbf{B} \rightarrow \mathbf{R}^O$.

Axiom 1. Feasibility of axioms: $R_{bar}^* \in F$.

Axiom 2. Individual rational axiom: $R_{bar} \geq R_{NE}^*, \forall o$.

Axiom 3. Pareto efficiency axiom: $R_{bar} \geq R_{bar}^* = \psi(F, \Gamma) \Rightarrow R_{bar} = R_{bar}^* = \psi(F, \Gamma)$.

Axiom 2 shows that in addition to the bargaining solution there is no other way for Pareto improvement. We can derive the following axiom.

Axiom 4. The invariance of the linear transformation axiom: $R_{bar}^* \in F$; for any linear transformation ϕ , there is $\phi(\psi(F, \Gamma)) = \psi(\phi(F), \phi(\Gamma))$.

Axiom 5. Independence of the axiom of choice: to any convex set E there is a subset F , and it is given as follows:

$$\begin{aligned} R_{bar}^* \in E \subseteq F &\implies \\ \psi(E, \Gamma) &= \psi(F, \Gamma). \end{aligned} \quad (33)$$

Axiom 6. Symmetry axiom: for any (F, Γ) , there is $R_{MR}^{(i)*} = R_{MR}^{(j)*}, (R_{bar}^{(i)}, R_{bar}^{(j)}) \in F$. It means that, for any $(R_{bar}^{(i)}, R_{bar}^{(j)}) \in F$, $R_{bar}^{(i)*} = R_{bar}^{(j)*} \in \psi(F, \Gamma)$ must satisfy $R_{bar}^{(i)*} = R_{bar}^{(j)*}$.

Nash bargaining solution: Nash bargaining solution refers as the Nash equilibrium solution that the participants gain their interests in a game, when they have many times of bargaining and gradually eliminating the Nash solution.

Introduction of Pareto evolutionary method is to obtain the optimal solution of BGRAA.

We give a multiobjective optimization problem and make a feasible solution set as the optimization goal. If there is a feasible solution, for each solution, it would be either the global optimal solution or the global extreme value point.

Evolving any two different individuals in population, if there is dominance, it must meet the following two conditions: there is at least one target and the subgoals are not bad.

Theorem 6. Meeting axioms 1 to 6, the only solution of bargaining problem is the utility of vector that maximizes the product Nash bargaining solution. Namely, solutions meet the following constrained optimization problem:

$$\begin{aligned} \psi(\mathbf{F}, \Gamma) \in \arg \max \quad & \prod_{o=1}^O (R_{bar}^{(o)*} - R_{MR}^{(o)*}) \\ \text{s.t.} \quad & R_{bar}^* \in \mathbf{F} \\ & R_{bar}^{(o)*} \geq R_{MR}^{(o)*}. \end{aligned} \quad (34)$$

Considering the resource of resource provider s^{\max} and the limitation of lowest income of resource provider in non-cooperative game, the optimization goal is to give the effective control to the resource provider and achieve the goal of utility optimization. The selection of the utility function U is given as follows:

$$\begin{aligned} U = \prod_{o=1}^O (R_{bar}^{(o)*} - R_{MR}^{(o)*}) \\ \text{s.t.} \quad R_{bar}^* \in \mathbf{F} \\ R_{bar}^{(o)*} \geq R_{MR}^{(o)*} \\ s_o \leq s^{\max}. \end{aligned} \quad (35)$$

5.2. Solution for Bargaining Game Problem of BGRAA. The solution of (35) is needed to be divided into two kinds of solution as follows.

- (a) Nash solution for bargaining for two people in the game is considered.

When $o = 2$, (35) can be represented as

$$\begin{aligned} (R_{bar}^{(i)*}, R_{bar}^{(j)*}) = \psi(\mathbf{F}, \Gamma) \\ = \arg \max (R_{bar}^{(i)} - R_{MR}^{(i)*}) (R_{bar}^{(j)} - R_{MR}^{(j)*}). \end{aligned} \quad (36)$$

The resources provisioning algorithm when $o = 2$ is as follows:

Step 1. Initialize the parameters $s_i = s_i^{\text{init}}$, $s_j = s_j^{\text{init}}$.

Step 2. Select the increment step length I , where $s_i^{(k+1)} = s_i^{(k)} + I$, $s_j^{(k+1)} = s_j^{(k)} + I$, and calculate the Pareto optimal point between $[0, s_i^{\max}]$, $[0, s_j^{\max}]$.

Step 3. From $(R_{bar}^{(i)} - R_{MR}^{(i)*}) \times (R_{bar}^{(j)} - R_{MR}^{(j)*}) = C$, calculate the corresponding strategy s_i , s_j , and the earnings of resource providers in NBS.

IF s_i and s_j reach convergence **THEN**

Return to Step 2.

ELSE

Record each convergence value. s_i and s_j are the optimal solution of the system.

Or calculate content conditions: $\pi = \arg \max((R_{bar}^{(i)} + R_{bar}^{(j)}) / ((s_i + s_j) | R_{bar}^{(i)} - R_{bar}^{(j)} |))$, then return s_i and s_j .

ENDIF

- (b) Nash solution for bargaining when the number of people is o in the game.

By Hungary algorithm, the problem can be converted into $o = 2$ bargaining problems, so as to provide the optimal system resources. Bargaining algorithm is shown as follows.

Step 1. Obtain the parameters from the provider and demand ones in the cloud market environment.

Step 2.

IF the number of the resource provider is even number **THEN**

Go to the Step 3.

ENDIF

IF the number of the resource provider is odd number **THEN**

Give an provider with none resource.

ENDIF

Step 3. Using the Hungarian method to make the resource provider into two groups.

Step 4. Through bargaining algorithm, return to the two bargaining Nash standard solution, and compute s_i , s_j , $R_{bar}^{(i)}$, $R_{bar}^{(j)}$.

Step 5. Go to Steps 3 and 4, until the NBS cannot be effective, return the parameters: s_i , s_j , $R_{bar}^{(i)}$, $R_{bar}^{(j)}$.

6. Performance Evaluation

Considering three types of resources, the CPU, memory, and hard disk, we use Google Cluster Trace workload reference data source as the data center, including user information, task types, task number, the number of subtasks, the subtasks resource request parameters, and resource utilization. Local clusters are composed of four computers and each computer is configured to CPU Intel (R) Core (TM) 2 Duo E8200 2.66 GHz, 3 GB memory, and 160 GB hard drive. These nodes are interconnected by a 1 Gbps bandwidth switch. Amazon EC2 environment consists of 80 small instances, which are configured to Inter Xeon CPU E5430 2.66 GHz, 1.7 GB memory, and 146.77 GB hard drive.

First, we show the performance of the algorithm itself. Then, we compare the performance with HM (Hadoop

Mechanism) and GM (Google Management Mechanism) individually, in the aspect of fair distribution, average utility, and the system throughput. Finally, we compare the algorithms NCGRAA, BGRAA, and GEA (General Equilibrium Algorithm) at the average utility, system throughput, and fairness.

In simulation, considering the length of the buffer and the queue, when $n = 100$, $n = 200$, and $n = 400$, the diversification of average queue length in the system is shown in Figure 2.

In Figure 2, in $M/M/50/50 + l$ model, the average queue length curve with increasing buffer changes smoothly. In $M/M/100/100 + l$, then, the average queue length slowly increases. In $M/M/200/200 + l$, the change of the buffer size does not affect the average queue length. From Figure 2, we can provide a buffer to ease the pressure on scarce resources in the system, while it will not cause big losses. In terms of shared resources in cloud computing system, equitable distribution of resources is especially important. Thus, the establishment of a buffer can reduce the load on the system to some extent, and it improves the overall performance of the system.

Figure 3 shows the influence of different task arrival rate on resource consumer revenue.

Task arrival rate is defined as the number of tasks when tasks arrive per minute. For fixed task arrival rate, the revenue increases with an increasing number of VMs. The lesser the request of the number of VMs, the longer the waiting time and the higher the loss for users. Thus performance degradation speed is fast. As the request of the number of VMs increases, the price of resource consumers paying the provider will also increase. The change of revenue is from increase to decrease; at this situation, the optimal number of VMs can maximize resources consumer revenue.

We set different number of resource providers, to make a research on the influence between the different size of the market resources and the convergence property. We give the number of resource providers from 3 to 6. With these numbers of resources providers, Figures 4(a) and 4(b) show how the utility changes with the number of iterations at the NCGRAA algorithm.

In Figures 4(a) and 4(b), initially, the resource providers constantly adjust strategy, to seek the best Nash equilibrium solution for resources provided. Subsequently, utility tends to be steady, and the resources reach equilibrium. At the same time, the lesser the number of resource providers is, the faster the NCGRAA algorithm can converge. As the number of resource providers increases, more iteration is needed to achieve utility state of stability. The overall situation is that the strategy can get a good convergence.

The results above indicate the robustness and convergence of the NCGRAA algorithm. In convergence condition, the number of the VMs to provide is the focus problem in Nash equilibrium strategy for resource provider; therefore, the cooperation in the market resource provider can timely adjust the strategy to achieve Nash equilibrium.

The third set of experiments investigates the performance of several algorithms, including the average utility, system throughput, and fairness.

Figures 5(a) and 5(b) show the changes of utility with BGRAA algorithm under different size of the market, the initial parameters, and size of the market that is the same to NCGRAA algorithm's.

It can be seen in Figures 5(a) and 5(b) that, with the increase of number of resource providers, more iteration is needed to achieve Nash bargaining equilibrium strategies. Namely, the Nash bargaining solution is in the Pareto optimal set which makes the products of the utility maximize. When utility has reached the convergence, then Pareto optimal solution reaches the border part of the negotiation strategy sets. At this point, the corresponding negotiation strategies and the corresponding utility for all resource providers are fair.

Figure 6 shows the influence of the amount of resources consumers on the average utility of the resource provider. It can be seen in Figure 6, with the increase of number of consumers, that utility increases. This is due to the increase of resource request. In addition, for the same resource consumer number, the effectiveness of the algorithm BGRAA is always higher than or equal to the utility of NCGRAA algorithm. This is because in bargaining negotiation among resources providers can reduce the number of VMs but enhance the utility of resource prices. In the market competition, resource providers compete against each other for more VMs. The disadvantage of this issue is that the resource price is low. Then the overall effect cannot achieve the maximum, but it can achieve the optimal value. Each HM or GM adjusts resource prices, based on general equilibrium without considering the relationship between competition and cooperation. Thus the average utility of HM and GM is lower than the ones of NCGRAA and BGRAA.

Figure 7 shows the change of system throughput. It can be seen in Figure 7 that as the number of resource providers increases the number of tasks per unit time also increases. Therefore, the system throughput has increased continuously. The throughput of BGRAA algorithm is the biggest of these algorithms by reducing resource price offering more VMs. In NCGRAA algorithm, when the total utility increases at the same time, the throughput increases. Although the HM and GM without the game theory are effective, they are not suitable for the system with some market parameters.

Figure 8 shows the fairness of the comparison results. The fairness of the allocation of resources in the cloud computing is significantly important. In BGRAA algorithm, performance parameter π is related to fairness and the throughput. It can be seen from Figure 8, along with the increase of the number of resource providers, that the change trend of HM and GM is bigger than that of NCGRAA and BGRAA algorithms. HM and GM algorithms ignore the mutual relationship between resource providers and resource allocation. The NBS of BGRAA algorithm encourages devotion to their utility. In NCGRAA algorithm, the utility is optimal and fairness principle is stable.

As shown in Figure 9, it can be seen that with the increasing number of consumers utility increases, because of the number of requested resources. As can be seen from Figure 9, the effectiveness of the algorithm is always higher than or equal to the utility algorithms. This is

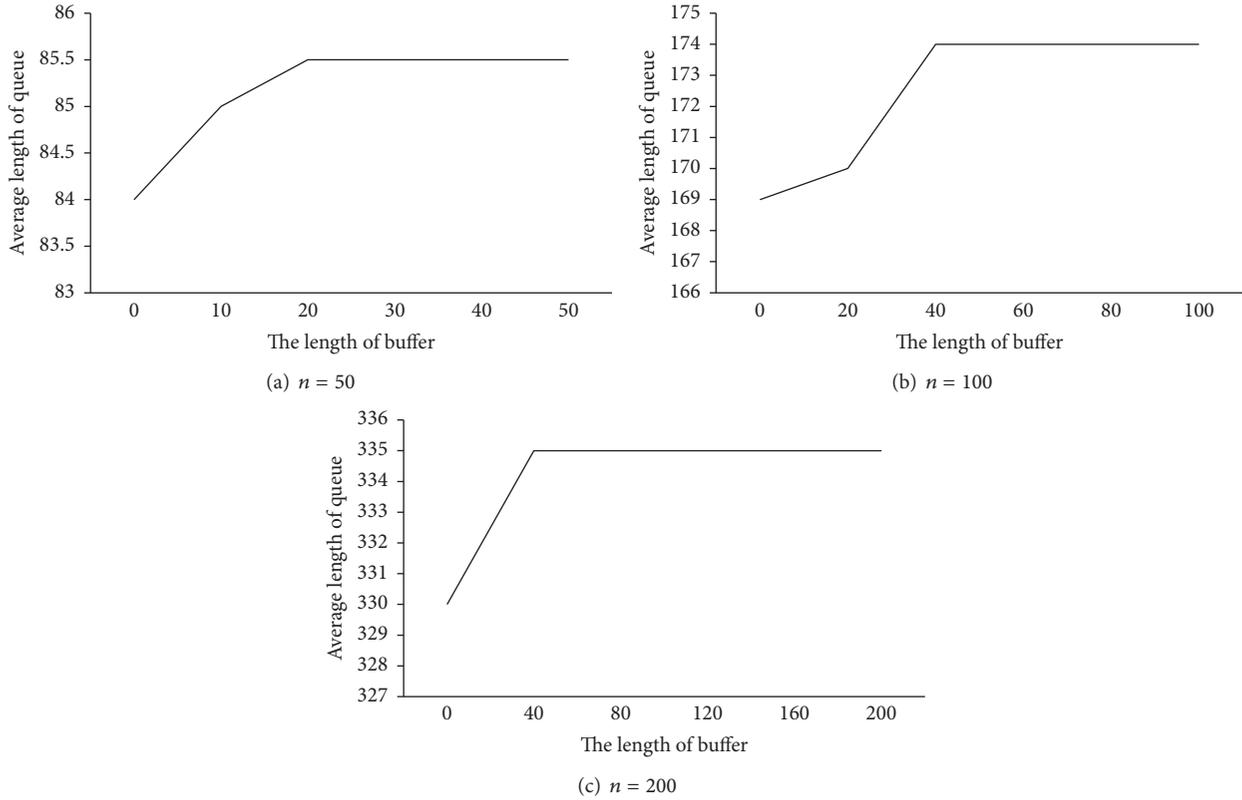


FIGURE 2: Influence of l on the average of length of queue.

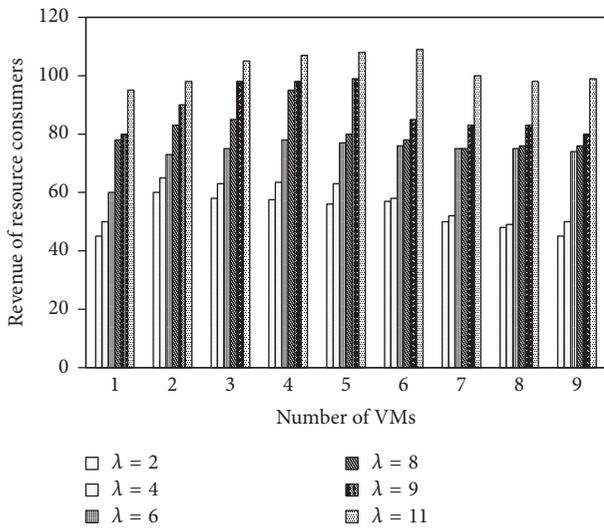


FIGURE 3: Influence between the number of VMs and revenue of resource consumers.

because in the negotiable market consultations between the resource providers can reduce the number of VMs and reach higher resource prices and utility. In noncooperative market competition, the resource providers provide more VMs competing for more market sharing. However, with lower resource prices, the overall effectiveness only achieves mutual optimum. GEA algorithm does not consider competition

and cooperation between providers of resources. And the resources are adjusted according to consumers' needs. That is bounded to affect the individual income.

Figure 10 shows the changes in system throughput; it can be seen that with the increased number of resource providers the number of tasks processed per unit of time also increases, so system throughput increases. NCGRAA by reducing resource prices provides more VMs and obtains the maximum throughput. On the contrary, the total utility increases with the loss of some throughput. GEA's throughput improvement curve is lower than the algorithms without considering cooperation competition between resources and users.

Figure 11 shows a fair comparison of the three algorithms. In the algorithm BGRAA, the parameter π is involved in fairness and throughput. As can be seen from the figure, as the number of resource providers increases, trends of NCGRAA and BGRAA are significantly greater than the GEA, which ignores the interaction between resource providers and lacks fairness. The welfare of bargaining between the two sides attaches great importance. It does not encourage the two sides to pursue their own utility while ignoring the other utilities.

7. Conclusion

The high efficiency and stable resources mechanism is needed for study of cloud computing resources management. Therefore, NCGRAA and BGRAA algorithms were proposed in

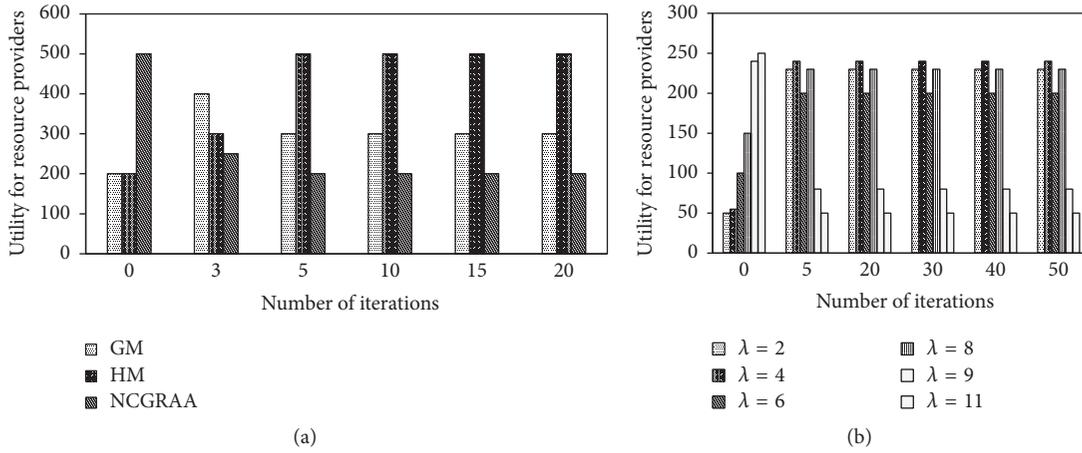


FIGURE 4: Influence on the utility with NCGRAA.

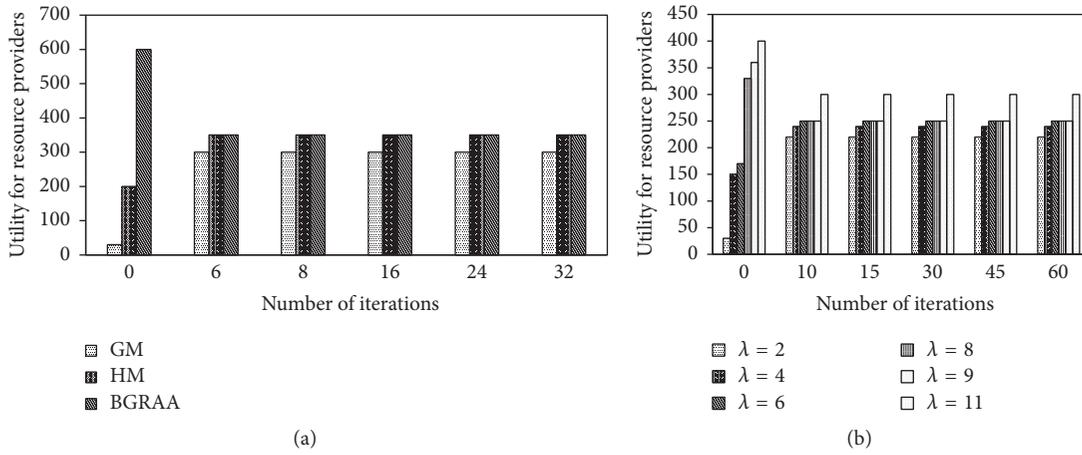


FIGURE 5: Influence on the utility with BGRRA.

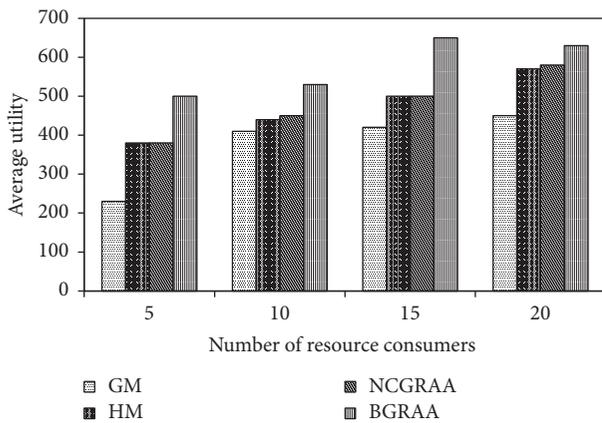


FIGURE 6: Influence of resource provider number on the system average utility.

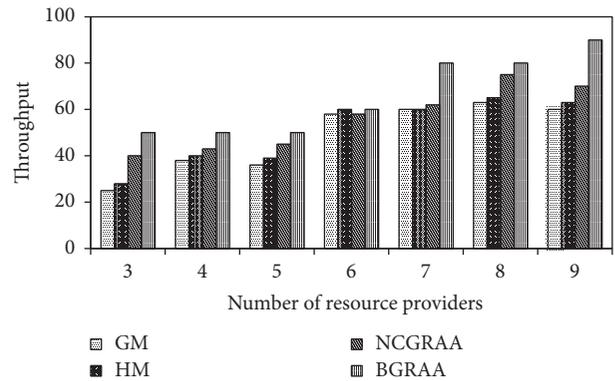


FIGURE 7: Influence of resource provider number on the system throughput.

this paper to solve this problem. The algorithms made competing resource allocation converge to the Nash equilibrium strategy. Meanwhile, algorithms bring resource competitors

optimal benefits and improved collective benefits. Based on the numerical results of some key performance measures, we discussed the influence of NCGRAA and BGRRA algorithms. These algorithms not only ensured fair constraints,

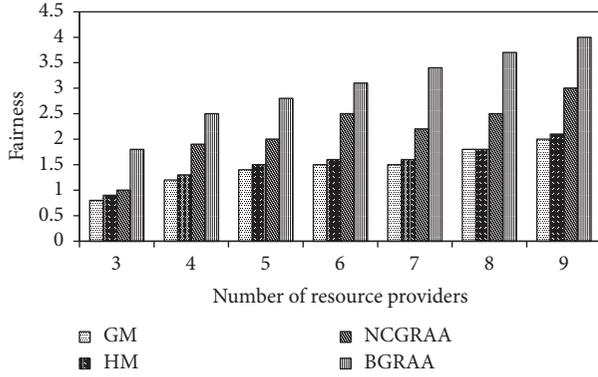


FIGURE 8: Influence of resource provider number on the fairness.

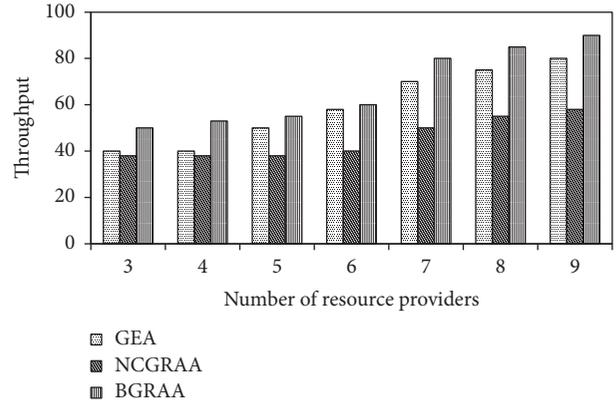


FIGURE 10: Comparison of throughput in different algorithms.

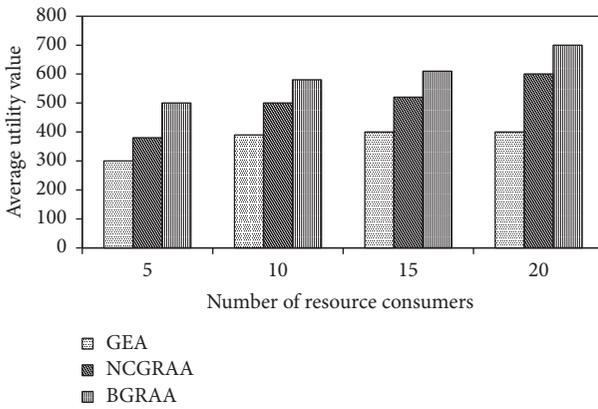


FIGURE 9: Comparison of average utility in different algorithms.

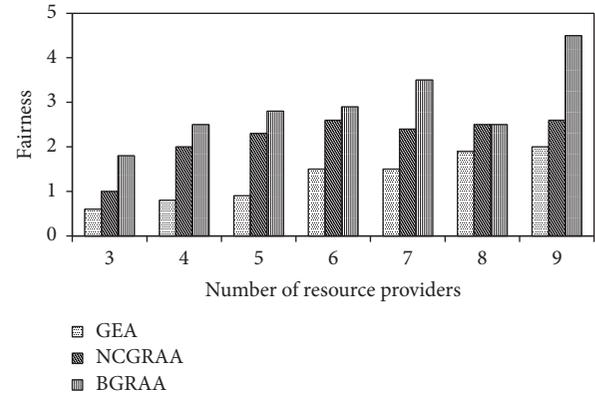


FIGURE 11: Comparison of fairness in different algorithms.

but also improved the efficiency of resource allocation. Meanwhile, it also shows that NCGRAA and BGRAA algorithms result in the slight increase of consumption.

In this work, we only focus on the resource allocation in the cloud computing. As a future work, we will extend the research by considering data transmission in other specific circumstances and security transaction in the negotiation process.

Parameters

λ :	Arrival rate of task
l :	Buffer length
π_k :	Probability of the queue length being k
$N(t)$:	Queue length at time t
$\bar{N}_a(t)$:	Average queue length
N :	Number of tasks
$W_a(t)$:	Waiting time
$W(t)$:	Response time.

Competing Interests

The authors declare that they have no competing interests.

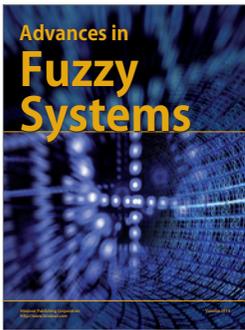
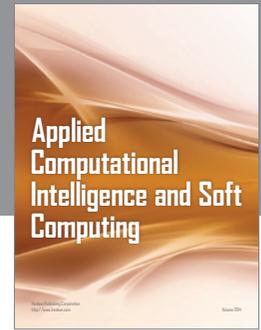
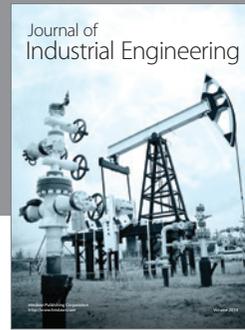
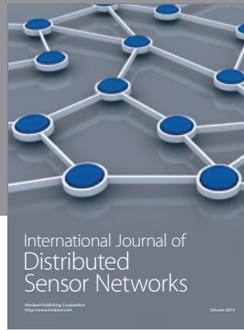
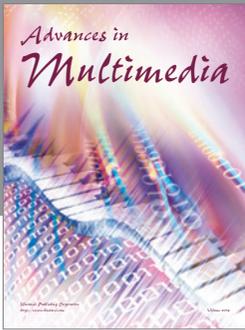
Acknowledgments

This research was supported by the Natural Science Foundation of China (no. 61300194).

References

- [1] J. Nim and A. Anandasivam, "SORMA-business cases for an open grid market: concept and implementation," in *Proceedings of the 5th International Workshop on Grid Economics and Business Models*, pp. 173–184, August 2003.
- [2] A. J. Younge and Z. Wang, "Efficient resource management for cloud computing environments," in *Proceedings of the Green Computing International Conference*, pp. 357–364, August 2010.
- [3] W. Yang, Z. Peng, L. Dong, Z. Hualiang, and Y. Haibin, "A performance modeling of decentralized cloud computing based on multiple $M/M/m/m+m$ queuing systems," *Acta Electronica Sinica*, vol. 42, no. 10, pp. 2055–2059, 2015.
- [4] G. Ping and B. Ling-ling, "The resource management model for cloud computing based on electronics," in *Proceedings of the Electrical & Electronics Engineering (EESYM '12)*, pp. 471–474, Kuala Lumpur, Malaysia, June 2012.
- [5] R. J. Al-Ali, K. Amin, G. V. Laszewski et al., "Scheduling independent multiprocessor tasks," *Journal of Grid Computing*, vol. 2, no. 2, pp. 163–182, 2004.
- [6] A. K. Amoura, E. Bampis, C. Kenyon, and Y. Manoussakis, "Scheduling independent multiprocessor tasks," *Algorithmica*, vol. 32, no. 2, pp. 247–261, 2002.

- [7] S. Borst, O. Boxma, J. F. Groote, and S. Mauw, "Task allocation in a multi-server system," *Journal of Scheduling*, vol. 6, no. 5, pp. 423–436, 2003.
- [8] R. Buyya, D. Abramson, J. Giddy, and H. Stockinger, "Economic models for resource management and scheduling in grid computing," *Concurrency and Computation: Practice and Experience*, vol. 14, no. 13–15, pp. 1507–1542, 2002.
- [9] A. Doğan and F. Özgüner, "Scheduling independent tasks with QoS requirements in grid computing with time-varying resource prices," in *Proceedings of the 3rd International Workshop on Grid Computing, Baltimore, Md, USA, November 2002*, vol. 2536, pp. 58–69, Springer, 2002.
- [10] M. A. Iverson, F. Özgüner, and L. Potter, "Statistical prediction of task execution times through analytic benchmarking for scheduling in a heterogeneous environment," *IEEE Transactions on Computers*, vol. 48, no. 12, pp. 1374–1379, 1999.
- [11] L. Keqin, "Experimental performance evaluation of job scheduling and processor allocation algorithms for grid computing on metacomputers," in *Proceedings of the 18th International Parallel and Distributed Processing Symposium (IPDPS '04)*, pp. 170–177, Santa Fe, NM, USA, April 2004.
- [12] K. Ranganathan, M. Ripeanu, A. Sarin, and I. Foster, "Incentive mechanisms for large collaborative resource sharing," in *Proceedings of the IEEE International Symposium on Cluster Computing and the Grid (CCGrid '04)*, pp. 1–8, IEEE, Chicago, Ill, USA, April 2004.
- [13] R. Wolski, J. S. Plank, T. Bryan, and J. Brevik, "G-commerce: market formulations controlling resource allocation on the computational grid," in *Proceedings of the 15th International Parallel and Distributed Processing Symposium*, pp. 46–52, April 2000.
- [14] J. K. Lenstra, D. B. Shmoys, and É. Tardos, "Approximation algorithms for scheduling unrelated parallel machines," *Mathematical Programming*, vol. 46, no. 1, pp. 259–271, 1990.
- [15] G. Christodoulou, E. Koutsoupias, and A. Vidali, "A lower bound for scheduling mechanisms," *Algorithmica*, vol. 55, no. 4, pp. 729–740, 2009.
- [16] C. Askarian, "A survey for load balancing in mobile WiMAX networks," *Advanced Computing*, vol. 3, no. 2, pp. 119–137, 2012.
- [17] D. Gkantsidis, O. Vytiniotis, D. Hodson, F. Narayanan Dinu, and A. Rowstron, "Rhea: automatic filtering for unstructured cloud storage," in *Proceedings of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI '13)*, vol. 106, no. 40, pp. 343–355, Lombard, Ill, USA, 2013.
- [18] Y. Dongmei and L. Chenghua, "Optimized collaborative filtering recommendation based on users' interest degree and feature," *Application Research of Computers*, vol. 29, no. 2, pp. 497–500, 2012.
- [19] H. Guodong, Z. Yige, and Z. Fan, "A dynamic replica placement approach based on cognition," *Computer Applications and Soft*, vol. 30, no. 1, pp. 83–87, 2013.
- [20] L. Wang, J. Luo, J. Shen, and F. Dong, "Cost and time aware ant colony algorithm for data replica in alpha magnetic spectrometer experiment," in *Proceedings of the IEEE International Congress on Big Data (BigData Congress '13)*, pp. 247–254, Santa Clara, Calif, USA, June 2013.
- [21] D. Yuan, Y. Yang, X. Liu et al., "A highly practical approach toward achieving minimum data sets storage cost in the cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1234–1244, 2013.
- [22] Dong and M. Wang, *Game Theory*, National Development and Reform Commission, Beijing, China, 2nd edition, 2008.
- [23] Y.-K. Kwok, K. Hwang, and S. Song, "Selfish grids: game-theoretic modeling and NAS/PSA benchmark evaluation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 5, pp. 621–636, 2007.
- [24] A. H. Elghirani, R. Subrata, and A. Y. Zomaya, "A proactive non-cooperative game-theoretic framework for data replication in data grids," in *Proceedings of the IEEE International Symposium on Cluster Computing and the Grid*, pp. 433–440, May 2008.
- [25] P. Ghosh, N. Roy, S. K. Das, and K. Basu, "A pricing strategy for job allocation in mobile grids using a non-cooperative bargaining theory framework," *Journal of Parallel and Distributed Computing*, vol. 65, no. 11, pp. 1366–1383, 2005.
- [26] C.-L. Li, D. Liao, L. Xiong, and Y.-J. Huang, "A service selection algorithm based on quantified QoE evaluation," *Acta Electronica Sinica*, vol. 43, no. 11, pp. 2145–2150, 2015.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

