Group Participation Game Strategy for Resource Allocation in Cloud Computing

Weifeng Sun[1], Danchuang Zhang[2], Ning Zhang[1], Qingqing Zhang[1], and Tie Qiu[1,*]

[1] School of Software, Dalian University of Technology
116621 Dalian Liaoning, China
{wfsun,qiutie}@dlut.edu.cn, zhang_ning@mail.dlut.edu.cn,
zhang901140@163.com

[2] Meteorological Administration of Dalian
116621 Dalian Liaoning, China
zhangdanchuang@163.com

Abstract. Based on the characteristics of cloud—resources belonging to the same institution and independent resource pool, we proposed a model for the complex task-resource and task-task interactions in cloud by game theory, and proved the existence of Nash equilibrium in the game. In this game model, every task selects resources by itself, rather than the resources are allocated by cloud system. We propose two cloud resource allocation game models—CT-RAG and CS-RAG. A new cloud resource allocation strategy—Group Participation Game Strategy (GPGS) is proposed based on these two game models. We also find out and analyze the equilibrium state of the game with GPGS. The theory analysis shows that GPGS can reduce the total cost of the system in the condition that all tasks/subtasks are rational. Simulation compares Nash, GPGS, Opt and "Round-Robin". The results of evaluation show that the GPGS is better.

Keywords: Cloud computing, resource allocation, game theory, Nash equilibrium.

1 Introduction

Recently, the cloud computing[1] has brought another innovation in IT industry. Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction[2]. Cloud computing can now be delivered as services over the internet. For example, CloudCast[3] provides the short-term weather forecasts depending on cloud computing services. The superiority of cloud computing compared to other parallel computing is the concept of resource pool. The resource pool is a virtual set of resources with discretionary combination and allocation, which can be expanded, allocated and recycled dynamically. The resources of cloud are diverse, like compute,

[*] Corresponding author.

storage, bandwidth, and so on. The cloud providers rent the resources to users and charge by the time or other metrics. For example, for Google App Engine [4], the Stored Data is charged by per gigabytes monthly.

However, the present pricing mechanisms and resource allocation strategies are far from optimal. When cloud system is in a high load state, the existing algorithms (e.g. FIFO, Round-Robin etc.) can't achieve good performance. This problem will become more and more prominent with the increase of cloud users. Moreover, the cloud users aim to reduce the turnaround time and payment of the tasks, and the cloud providers aim to improve the performance of the whole system. Hence, resource allocation in cloud is a very important issue for not only the whole system but also every task.

It is generally accepted that game theory is effective to solve many issues in computer science, such as in grid resource allocation [5], node stimulation in wireless networks and P2P networks[6] [7].The basic elements of a game include: player, action, payoff, information, strategy, outcome and equilibrium. In practice, a game is defined to model the issue to be solved with the basic elements, and the purpose is to find the Nash Equilibrium by game analysis. Nash equilibrium is a stable state with strategy set of all players. John F. Nash has proved that every game has Nash equilibrium [8]. Consequently, game theory can be applied to solve many issues in computer science. In cloud, the system manages and schedules different sources with the form of resource pool by virtualization, and the tasks of different users compete for these resources. So it is reasonable to define a game for cloud resource allocation. By setting a series of reasonable allocation strategies, we can adjust the Nash equilibrium state of this game. And these may improve the performance and load balancing of system.

In this paper, we combine the cloud resource allocation with game theory, and propose a new resource allocation strategy in cloud. It can minimize the total cost by change the payoff of each player, analyze the process and the result of the equilibrium state, and prove the allocation strategy we proposed can reduce the total cost of the system.

The main contributions of this paper are:

- We propose a new cloud resource allocation game model and a new resource allocation strategy GPGS based on the overall interests assuming that all tasks are rational;
- We proved the existence of Nash equilibrium and GPGS equilibrium in GPGS;
- Analysis demonstrates that the GPGS equilibrium state achieved from GPGS has smaller total cost of the whole system.

2 Related Work

With the development of cloud, traditional resource allocation strategy is no longer suitable in cloud computing environment, [9] compare the difference of resource allocation between cloud computing and traditional environments. That is:

- Divide the independent resource pool, allocate and release different physical and virtual resources according to the need of tasks dynamically;
- Provide and release the resources flexibly and fast;
- Cloud system need to optimize the use of resource automatically.

Cloud computing and grid computing are similar. Game theory has been widely used in the research of grid resource allocation problems [10] [11].Combined with the characteristics of cloud computing resource allocation, game theory can also be applied to solve the problem of resource allocation in cloud computing. [12] proposed a cloud resource allocation strategy in continuous double auction framework based on Nash equilibrium which can allocation the resources in the cloud environment effectively. [13] proposed a cloud resource allocation strategy based on the evolutionary game, achieving a minimum cost by change the resource selection strategy individually. [14] proposed a market-based resource allocation strategy, experiments results show that this strategy could achieve the Nash equilibrium and a balance between supply and demand effectively.

There are few papers which considered the conflict between the rationality of each task and the total cost of the system. Usually, the allocation strategies of tasks in cloud environments aim at the overall optimal system cost in the process of resource selection. If a task is always considering itself to obtain a better payoff, then we call it rational task. Rational tasks can change their strategies to enhance their own payoffs, however raising the overall system costs at the same time. In this paper, on condition that all tasks/subtasks are rational, we present a new cloud resource allocation game model and proposed a new strategy based on the overall interests. The system can achieve the minimum total cost while all tasks select resources base on this strategy. And all tasks can obtain maximum payoff in the game.

3 Resource Allocation Game Model and Nash Equilibrium

In this section, a Cloud Task-Resource Allocation Games (CT-RAG) is proposed and defined. CT-RAG models the resource allocation in cloud computing and captures the task-resource and task-task interactions. Next, it shows the existence of Nash equilibrium and the defect in an instance of CT-RAG because of the task rationality. This paper considers the tasks which clients submit during some time as a batch. Therefore, the resource allocation strategy of CT-RAG is considered as a static scenario. We leave the real-time situation to future work.

3.1 Resource Allocation Game Modeling

In our model, the tasks act the player of a CT-RAG. The resource allocation determines the strategy of the tasks (players). In other words, task obtains the payoff by means of selecting different resources. In this paper, there are several assumptions about CT-RAG as follow:

1. Subtask is defined as the smallest execution unit in a cloud. Every task can be separated by several same or similar subtasks. And all the subtasks of different tasks have the same requirement of resource and execution time on a determinate resource. The subtasks of one task are independent. One task finished implies that all the subtasks of this task are finished.
2. There is only one kind of resource (e.g. CPU) in this model. The tasks or subtasks are charged base on execution time. It means that tasks or subtasks have to pay to

cloud provider based on the expense function. The execution time will increase with the increase of task number on one resource.

3. All tasks or subtasks are rational. It means that the only thing tasks considered is how to increase their own payoff, rather than how to increase the system's payoff.
4. The cost of one task is composition of execution time and the expense of price charged by resource. The execution time of a task is the sum of all subtasks' execution time. In this work, the cost of one task is a sum with the weight of execution time and expense. And we defined the ratio is 1:1.

$$cost = completion\ time + expense\ . \tag{1}$$

Suppose there are m tasks $S= (S_1, S_2, ..., S_m)$ and n resources $R= (R_1, R_2, ..., R_n)$ in the cloud. All the tasks and subtasks simultaneously use the resources of cloud. There is no limit to the amount of subtasks executed on one resource. For each subtask, the execution time would enhance with the increase in amount of subtasks executed on the same resource.

Each task $S_i \in S$ can be divided into x_i subtasks, therefore the task-resource interactions can be considered as task-task interactions. Each subtask can only execute on one resource. Assume that the amount of all subtasks is M, which is $\sum_{i=1}^{m} x_i=M$. Each resource $R_j \in R$ is associated with an expense function $f_j(\cdot)$ $(f_j \in IR_+)$ and single execution time t_j of each subtask. $f_j(y)$ is the expense of every subtask when there are y subtasks executed on resource R_j. This paper assumes that f_j is a monotonic increasing function and $y \cdot f_j$ is also a monotonic increasing function. The vector $f = (f_1, f_2, ..., f_n)$ shows the expense functions of all the resources in the cloud. By changing the expense function of each resource, various resources can be distinguished in the cloud. Such as the increase of f_j' can result in the decrease of y_j, and y_j is the amount of subtask executed on resource R_j at Nash equilibrium. Obviously,$\sum_{i=1}^{m} x_i=\sum_{j=1}^{n} y_j=M$ is obtainable. The costs (completion time and expense) of each subtask executed on the same resource are consistent. The vector $t = (t_1, t_2, ..., t_n)$ denotes the execution time of all the resources in the case of one resource only executes one subtask. The completion time of each subtask on resource R_j equals the product of the execution time t_j and y_j of this resource, which is $T_j= y_j \times t_j$.

The expense function of each resource represents an abstraction of expense when the task or subtask is executing on the resource, and is proportional to the ability of the resource. All tasks must pay the money for the execution on the selected resources, this is where expense occurs. On the condition that the cloud defines the expense function of each resource according to its ability, the priority of each task can be distinguished by given them different amount of money. But in our paper, there is no limit of the amount of money for each task. We leave the discussion of it to future work.

In this paper, the allocation interaction between tasks and resources is represented as a global resource allocation matrix A. Matrix A is composed of m rows, one for each task, and n columns, one for each resource (Fig. 1(a)). $a_{ij}=q$ indicates that q subtasks of task S_i are allocated to resource R_j, and $a_{ij}=0$ indicates there are not any subtasks allocated to resource R_j. Global resource allocation matrix A is also regarded as a m-dimensional vector $(a_1, a_2, ..., a_m)^T$. a_i is the strategy adopted by task S_i in CT-RAG and is given by the vector $(a_{i1}, a_{i2}, ..., a_{in})$. The feasibility of the global resource allocation matrix A depends on the following conditions:(1) $\sum_{j=1}^{n} a_{ij}=x_i$ and

(2) $a_{ij}\geq 0\ \forall i,j$. Similarly, there is a subtask-resource allocation matrix B. It can reflect the allocation interaction between subtasks and resources (Fig. 1 (b)). In CT-RAG, each subtask can only execute on one resource, so b_{ij} can only be two values $b_{ij}=0$ or $b_{ij}=1$ and there is only one "1" in each row (strategy of each subtask).

The 3-tuple (S, A, P) represents a Cloud Task-Resource Allocation Game (CT-RAG). In CT-RAG, the player is task, and the strategy of S_i is the i-th row of matrix A— (a_{i1}, a_{i2}, ..., a_{in}). The vector P= (P_1, P_2, ..., P_m) is the payoff of all players (tasks), and P_i is the reciprocal of the cost C_i of task S_i. P_i and C_i is shown as Formula (2). And the performance of the whole system is reflected by the total cost C which is the sum of the cost of all tasks according to Formula (2).

$$A=\begin{array}{cc}\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n}\\ a_{21} & a_{22} & \cdots & a_{2n}\\ \vdots & \vdots & \ddots & \vdots\\ a_{m1} & a_{m2} & \cdots & a_{mn}\end{pmatrix} & \begin{matrix} x_1\\ x_2\\ \vdots\\ x_m\end{matrix}\\ \begin{matrix} y_1 & y_2 & \cdots & y_n\end{matrix} & M\end{array} \quad B=\begin{array}{cc}\begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n}\\ b_{21} & b_{22} & \cdots & b_{2n}\\ \vdots & \vdots & \ddots & \vdots\\ b_{M1} & b_{M2} & \cdots & b_{Mn}\end{pmatrix} & \begin{matrix} 1\\ 1\\ \vdots\\ 1\end{matrix}\\ \begin{matrix} y_1 & y_2 & \cdots & y_n\end{matrix} & M\end{array}$$

(a) (b)

Fig. 1. Allocation matrixes of tasks-resources and subtasks-resources: (a) tasks-resources matrix **A** ;(b)subtasks-resources matrix B

$$\begin{aligned} C_i &= \sum_{j=1}^{n} a_{ij}\times(T_j\times y_j + f_j(y_j))\\ P_i &= 1/C_i = 1\Big/\sum_{j=1}^{n} a_{ij}\times(T_j\times y_j + f_j(y_j))\\ C &= \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}\times(T_j\times y_j + f_j(y_j)) \end{aligned} \quad . \tag{2}$$

The cloud system aims to minimize the total cost C, and every task also tries to decrease its own cost C_i by altering the strategy.

3.2 Nash Equilibrium in CT-RAG

CT-RAG has a Nash equilibrium certainly like other games. In CT-RAG, each task always tries to reduce its own cost by altering strategy continually according to the strategies of others. This dynamic process with continuous changing can't be stop until the system is in a stable state. In other words, all tasks can't reduce its cost by changing its strategy. This state is the Nash equilibrium of CT-RAG.

Theorem 1: If all the expense functions of CT-RAG are linear, the Nash equilibrium of CT-RAG is not unique. And there is the same z_j, $z_j=\sum_{i=1}^{m} a_{ij}$, in all Nash equilibriums.

Proof: In Nash equilibrium of CT-RAG, no task has motivation to change the stable state. It means the cost of subtask executed on each resource is the same. If a new subtask need to execute, its cost is the same whichever resource it selected. Assume that the amount of subtask executed on each resource is (z_1, z_2, …, z_n). This can be express as follow for each resource:

$$\begin{cases} t_1(z_1+1)+f_1(z_1+1)=\cdots=t_n(z_n+1)+f_n(z_n+1) \\ \sum_{j=1}^{n} z_j = M \end{cases} \tag{3}$$

Since t_i, f_i, M is given, the homogeneous linear system consisting of n equations in n unknowns can be solved. There is the same z_j, $z_j=\sum_{i=1}^{m} a_{ij}$, in all Nash equilibriums. Therefore, a resource allocation matrix A is Nash equilibrium if it satisfies the following properties: (1) $\sum_{j=1}^{n} a_{ij}=x_i$, (2) $\sum_{i=1}^{m} a_{ij}=z_j$ and (3) $a_{ij}\geq 0$ $\forall i,j$. So the Nash equilibrium of CT-RAG is not unique.

Theorem2. If CT-RAG has linear expense function then the total cost of any Nash equilibrium is at most 4/3 times as much as the total cost of theory optimal [15].

The proof of theorem 2 is shown in [16]. The gap of the two total cost is caused by the rationality of tasks. In most instances, the strategy change of one task can result in the cost increase of other tasks and the total cost C.

4 The Group Participation Game Strategy

To reduce the influence mentioned in section 3, we proposed a new resource allocation Group Participation Game Strategy (GPGS). Every task/subtask selects its strategy according to GPGS by itself. It ensures the cost of the whole system at Nash equilibrium is close to the global optimum. In this section, we assume the expense function is linear. Other cases of expense function exceed the scope of this paper, and linear expense function is feasible for cloud to adjust the Nash equilibrium. In this section, a Cloud Subtask-Resource Allocation Games (CS-RAGs) is proposed and defined firstly. The players of CS-RAGs are subtasks, rather than tasks. The analysis of total cost belongs to CT-RAGs is similar as that of CS-RAGs, so we can calculate the amount of subtasks executed on each resource (Y_1, Y_2, …, Y_n). Then all tasks select the resources in order according to this vector.

4.1 CS-RAGs and Nash Equilibrium

The 3-tuple (s, B, p) represents a Cloud Subtask-Resource Allocation Game (CS-RAG). CS-RAG is a sequential game, and every subtask selects its strategy one by one. In CS-RAG, the player is subtask. s_i represents the i-th subtask, and the strategy of s_i is the i-th row of matrix B— (b_{i1}, b_{i2}, ..., b_{in}). Every subtask can only be allocated to one resource, so the strategy of s_i is also represented as w_i, $w_i \in Z$ and $1 \leqslant w_i \leqslant n$, it means the subtask has been allocated to resource R_{w_i}. The vector $p=$ (p_1, p_2, ..., p_M) is the payoff of all players (subtasks), and p is the reciprocal of the cost c_i of subtask s_i. c_i and the total cost C is shown as follow:

$$\begin{aligned} c_i &= T_{w_i} \times y_{w_i} + f_{w_i}(y_{w_i}) \\ C &= \sum_{i=1}^{M} c_i = \sum_{i=1}^{M}\left[T_{w_i} \times y_{w_i} + f_{w_i}(y_{w_i})\right] \end{aligned} \tag{4}$$

And the payoff of every subtask is defined as follow:

$$p_i = 1/c_i = 1/T_{w_i} \times y_{w_i} + f_{w_i}(y_{w_i}) \quad . \tag{5}$$

As the analysis above, CS-RAG also has Nash equilibrium, and it is not unique. We can get the number of subtask on every resource in Nash equilibrium of the game with M subtasks and n resources according to formula (3), (4), (5).

$$Y_j' = (M - \sum_{h=1}^{n}((b_j - b_h)/(t_h + a_h)))/((t_j + a_j)\sum_{h=1}^{n}(1/t_h + a_h)) \quad . \tag{6}$$

4.2 Spillover Effect or Externality

Consider the situation that m tasks compete for two resources R_1 andR_2.Assuming that number of all subtasks is M. Each subtask has only two choices—R_1 or R_2. Strategy of subtask s_i depends on the strategy of other M-1 subtasks; and strategy of each subtask will have an impact on other subtasks, which is called spillover effect. Supposing there are y subtasks which choose R_1, the cost of each subtask is$c_1(y)$, and M-y subtasks which chooseR_2, the cost of each subtask is $c_2(y)$, then the total cost of the system is as Formula (7)

$$C(y) = yc_1(y) + (M - y)c_2(y) \quad . \tag{7}$$

If subtask s_i choose R_2 at beginning, and change the strategy to choose resource R_1, then the number of subtasks which choose R_1 rise to y+1.The cost of s_i change from $c_2(y)$to $c_1(y+1)$, and the total cost of the system is as Formula (8).

$$C(y+1) = (y+1)c_1(y+1) + (M - y - 1)c_2(M - y - 1) \quad . \tag{8}$$

The difference between $C(y)$ and $C(y$+1) is the increment of total payoff.

$$C(y+1) - C(y) = (c_1(y+1) - c_2(M-y)) + y(c_1(y+1) - c_1(y)) + (M-y-1)(c_2(M-y-1) - c_2(M-y)) \quad . \tag{9}$$

The first item$c_1(y+1)$-$c_2(M-y)$in Formula (9) is the payoff alteration of strategy changer, called the marginal private gain. When the change of one subtask affects other subtasks, the payoff alteration of other subtasks is called marginal spillover effect, which is the second and third item of Formula (9). The spillover effect leads to the payoff conflict between tasks and the whole system.

4.3 The Group Participation Game Strategy (GPGS)

Because of the subtask rationality, Nash equilibrium of CS-RAG can't minimize the total cost. Thus this paper proposes a group participation game strategy (GPGS) to adjust CS-RAG, and all players in this game must observe this strategy. There are three regulations of resource allocation strategy GPGS:

Regulation 1: x_i is subtask number of task S_i. Task S_1, S_2, ..., S_m are sequenced according to x_i. Assuming that S_1, S_2, ... , S_m is a sequential vector and $S_1$is the task with least subtasks. The regulation of resource selecting for all tasks is: S_1 selects x_1 resources firstly. The next is S_2, and S_m is the last one. All the subtasks will be executed synchronously when all tasks have already selected resources.

Regulation 2: $c_1, c_2, \ldots, c_n$ are the cost of subtask on each resource. Every task selects resource according to the vector $c_1, c_2, \ldots, c_n$, and it can be calculated by this:

$$c_j = Y_j \times t_j + f_j(Y_j) \quad (10)$$

In this paper, $f_j(\cdot)$ is linear, so c_j can also be expressed as follow:

$$c_j = Y_j \times t_j + (a_j \times Y_j + b_j) \quad (11)$$

Regulation 3: The payoff of every subtask is defined as follow:

$$p_i = 1/(c_i + \lambda c_\Delta) \quad (12)$$

When some subtask is selecting resource, λ is the number of subtask which has already selected this resource, and c_Δ is the increment of every subtask on this resource.

In practice, tasks select resources according to these three regulations, and it can be seen as a CT-RAG with GPGS. The game with GPGS will converge to equilibrium state. And we call it GPGS equilibrium in this paper. We attempt to ensure the total cost of the GPGS equilibrium is close to optimal.

4.4 The Theory Analysis of GPGS

The aim of cloud computing resource allocation is achieving the least overall cost. So the cloud computing resource allocation game can be regard as group game, and it is appropriate that analyze the whole interest of cloud resource allocation by group game theory. In group game, the strategy every player selected is not necessarily known. The group game analysis focus on the player number of each strategy. We can judge the performance of an allocation strategy by comparing the gap between the Nash equilibrium and the GPGS equilibrium.

In CT-RAG, task S_i can be divided into x_i subtasks, so it has $C_{x_i n}$ strategies. It is complicated to analyze the overall interest by the unit of task. In this paper, the overall interest of all tasks is the same as the overall interest of all the subtasks, and every subtask only has n strategies. Therefore, the overall interest of collective action can be discussed by the unit of subtask. Then the analysis of CT-RAG turns into the analysis of CS-RAG. First, spillover effect or externality will be described in CS-RAG. Next, an example of m tasks and two resources will be given in order to show the convergence process of Nash equilibrium. At last, the gap of the total cost between the Nash equilibrium and the GPGS equilibrium will be deduced with liner expense function by a general example. In this section, the increasing of overall interest is replaced by the decreasing the total cost.

We have already analyzed the Nash equilibrium of CS-RAG in section 4.1. The primary difference between CS-RAG and CS-RAG with GPGS is the definition of payoff. In CS-RAG with GPGS, the payoff is defined as the reciprocal of $c_i+\lambda c_\Delta$.

If some subtask intends to enhance its own payoff, it is necessary to reduce not only its cost, but also the impact of its strategy on other subtasks. Therefore, the rationality is combined with the interest of whole system.

The convergence process of GPGS equilibrium is almost the same as Nash equilibrium. Every subtask chooses the strategy which can maximize its own payoff. At the same time, it can minimize the whole cost. We can get it from the definition of payoff easily.

In the example of *m* tasks composed of *M* subtasks and two resources we mentioned in section 4.1, the equilibrium state will change with the increase of GPGS strategy in the game. The total cost can be represented by this:

$$\begin{aligned} C &= y\times c_1(y)+(M-y)\times c_2(y) \\ &= (t_1+a_1)\times y^2+b_1\times y+(t_2+a_2)\times(M-y)^2+b_2\times(M-y) \end{aligned} \quad (13)$$

The GPGS equilibrium is the state with minimum total cost. So we can get the value of *y*:

$$y=(M(t_2+a_2)+0.5(b_2-b_1))/(t_1+t_2+a_1+a_2) \quad (14)$$

We can calculate the total cost in the two states and the difference between them:

$$\begin{aligned} &C_{Nash}=M^2(t_2+a_2)+Mb_2-(M^2(t_2+a_2)^2+M(t_2+a_2)(b_2-b_1))/(t_1+t_2+a_1+a_2) \\ &C_{GPGS}=M^2(t_2+a_2)+Mb_2-(M^2(t_2+a_2)^2+M(t_2+a_2)(b_2-b_1)+(b_2-b_1)^2)/4(t_1+t_2+a_1+a_2) \\ &C_{Nash}-C_{GPGS}=(b_2-b_1)^2/4(t_1+t_2+a_1+a_2) \end{aligned} \quad (15)$$

Consider the general situation. Assuming there are *m* tasks composed of *M* subtasks and *n* resources. GPGS equilibrium is the state which can minimize the whole cost. And it can be shown as follow:

$$min\ C=\sum_{j=1}^{n}(Y_j''\times t_j+a_j\times Y_j''+b_j)\quad \sum_{j=1}^{n}Y_j''=M\quad 0\le y_j\le M,1\le j\le n \quad (16)$$

Where Y_j'' is the number of subtasks on resource R_j in GPGS equilibrium.

It can be solved by Lagrange Multipliers method. The process of the solution is shown as follow:

Introduce a function: If *C* is the minimum, it means. And we get a system of linear equations of *n* unknowns:

$$\begin{cases} 2(t_1+a_1)Y_1''+b_1+\lambda=0 \\ \qquad\vdots \\ 2(t_n+a_n)Y_n''+b_n+\lambda=0 \\ Y_1''+Y_2''+\cdots+Y_n''=M \end{cases} \quad (17)$$

Therefore, we can get the solution of the GPGS equilibrium:

$$Y_j''=(M-\sum_{h=1}^{n}((b_j-b_h)/2(t_h+a_h)))/((t_j+a_j)\sum_{h=1}^{n}1/(t_h+a_h)) \quad (18)$$

By the above analysis, there is a less total cost of the system in CS-RAG with GPGS, although all players aim to enhance their own payoff. In practice, the resource allocation scheme is every player select resource by itself according to GPGS, rather than the resource provider allocate integrally.

5 Experimental Evaluation

In this paper, we implement the evaluation test to verify the feasibility and performance of our resource allocation strategy. In cloud environment, resource usually consists of virtual resource pools and physical resources. These resources can be selected by each user dynamically as required. Physical host is the most common cloud computing node resource. In GPGS, resources are selected by each task based on group game to reduce the total system cost and improve the efficiency.

The aim of a resource allocation is obtain the best performance of the whole system. The total cost is an important index of the performance of the system. We compare the total cost of the Nash, Optimal, "Round-Robin" and GPGS by the model we proposed in section 3.1. "Round-Robin" has already been used well in system like Hadoop. In the "Round-Robin" method, all resources are numbered from 1 to *n*. when a task wants to execute with *k* resources, it is scheduled on the next *k* resources, as a Round-Robin manner.

In practice, there are sorts of resources in cloud. And we can distinguish these resources by different expense function in our model. We used two different examples to analyze different strategies:

- The capacities of all resources are similar. It means the expense function of them is similar too, and the functions are all linear.
- The gap between the capacities of different resources is large. They have obvious different linear expense function.

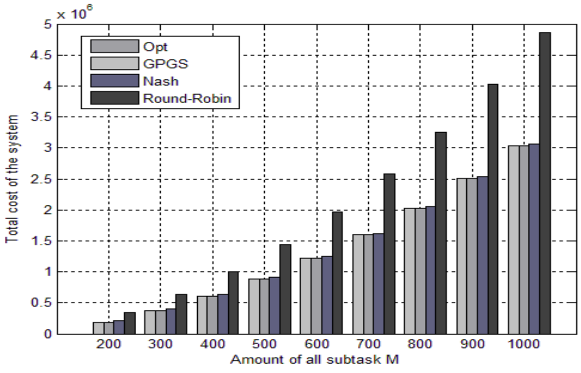

Fig. 2. The Comparison of total cost with homogeneous resources in different strategies

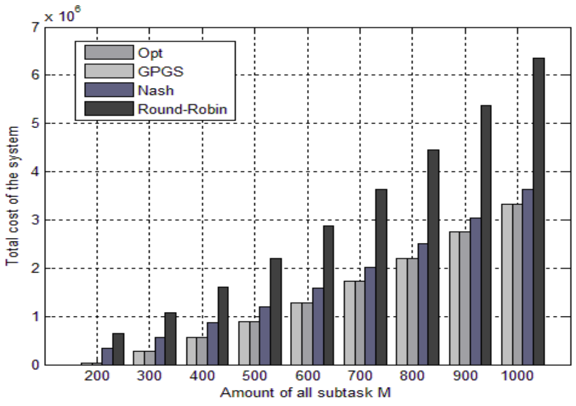

Fig. 3. The Comparison of total cost with heterogeneous resources in different strategies

For comparing the performance of different resource allocation strategies, we used five host resources with heterogeneous expense function. We recorded the total cost of each resource allocation strategies when the number of subtask is from 200 to 1000.Fig. 2 compares the performance of the various strategies with homogeneous resources, and Fig. 3 compares the performance of the various strategies with heterogeneous resources. In both two cases, the total cost of GPGS is less the "Round-Robin" obviously. When the expenses of different resources are quite different, the total cost of GPGS is less than Nash's, and it is quite close to Opt's.

6 Conclusion and Future Work

In this paper, we proposed a new resource allocation game model in cloud. Further, we demonstrated the existence of Nash equilibrium in the game and found that Nash equilibrium can't reduce the total cost. For reduce the total cost, we proposed a new cloud resource allocation strategy—group participation game strategy (GPGS), and compared the difference between Nash equilibrium state and GPGS equilibrium state in the game. By the analysis of the difference between the two states, we concluded that rational tasks lead to the increase of total cost. The gap between Nash equilibrium and GPGS equilibrium was shown by means of a special example. Simulations compared the total and load balance between Nash, Optimal, "Round-Robin" and GPGS. The result showed GPGS can reduce the total cost effectively.

However, limited to the page limit, we don't analyze the load balance and the task fairness deviation which represents the fairness of tasks. In addition, the game in this paper is static, and different subtasks are relative in many cases of cloud resource allocation. We do not take these into consideration in this paper. In future work, we will talk about the analysis the load balance and the task fairness deviation, and we will take the dynamic resource allocation strategy into consideration.

Acknowledgments. This work is supported in part by Natural Science Foundation of China under grant No. 61103233, 61202442, 61202443.

References

1. Jadeja, Y., Modi, K.: Cloud computing - concepts, architecture and challenges. In: International Conference on Computing, Electronics and Electrical Technologies, pp. 877–880 (2012)
2. Mell, P., Grance, T.: The NIST Definition of Cloud Computing. National Institute of Standards and Technology (2011)
3. Krishnappa, D.K., Irwin, D., Lyons, E., Zink, M.: CloudCast: Cloud computing for short-term mobile weather forecasts. In: IEEE International, Performance Computing and Communications Conference, pp. 61–70 (2012)
4. Google app engine, `http://appengine.google.com/`
5. Yaghoobi, M., Fanian, A., Khajemohammadi, H., Gulliver, T.A.: A non-cooperative game theory approach to optimize workflow scheduling in grid computing. In: Pacific Rim Conference on Communications, Computers and Signal Processing, Victoria BC, pp. 108–113 (2013)
6. Michiardi, P., Molva, R.: A collaborative reputation mechanism to enforce node cooperation in mobile ad-hoc networks. In: Proceedings of the IFIP TC6/TC11 6th Joint Working Conference on Communications and Multimedia Security, Deventer, The Netherlands, pp. 1072–1121 (2002)
7. Wang, T.-M., Lee, W.-T., Wu, T.-Y., Wei, H.-W., Lin, Y.-S.: New P2P Sharing Incentive Mechanism Based on Social Network and Game Theory. In: International Conference on Advanced Information Networking and Applications Workshops, Fukuoka, pp. 915–919 (2012)
8. Nash, J.: Non-cooperative Games. Annals of Mathematics 54, 289–295 (1951)
9. Foster, I., Zhao, Y., Raicu, I., Lu, S.Y.: Cloud Computing and Grid Computing 360-degree compared. In: Grid Computing Environments Workshop, Austin TX, pp. 1–10 (2008)
10. Li, Z.J., Cheng, C.T.: An Evolutionary Game Algorithm for Grid Resource Allocation under Bounded Rationality. Concurrency and Computation: Practice and Experience 9, 1205–1223 (2009)
11. Caramia, M., Giordani, S.: Resource allocation in grid computing:An economic model. WSEAS Transactions on Computer Research 3, 19–27 (2008)
12. Guiran, C., Chuan, W., Yu, X.: Efficient Nash Equilibrium Based Cloud Resource Allocation by Using a Continuous Double Auction. In: International Conferenceon Computer Design and Applications, Shenyang China, pp. 94–99 (2010)
13. Wei, G., Vasilakos, A.V., Zheng, Y., Xiong, N.: A game-theoretic method of fair resource allocation for cloud computing services. The Journal of Supercomputing 54, 252–269 (2010)
14. You, X.D., Wan, J.: ARAS-M: Automatic Resource Allocation Strategy based on Market Mechanism in Cloud Computing. Journal of Computers 6, 1287–1296 (2011)
15. Jalaparti, V., Nguyen, G.D., Gupta, I., Caesar, M.: Cloud Resource Allocation Games. Technical Report, University of Illinois (2010), `http://hdl.handle.net/2142/17427`
16. Roughgarden, T., Tardos, E.: How bad is selfish routing. Journal of the ACM 49, 236–259 (2002)