Computing and Informatics, Vol. 23, 2004, 189–203

MARKET MECHANISM FOR DYNAMIC RESOURCE MANAGEMENT IN COMPUTATIONAL GRID

Li ChunLin, Li LaYuan

Department of Computer Science, Wuhan University of Technology
Wuhan 430063, P. R. China
e-mail: chunlin74@163.net, jwtu@public.wh.hb.cn

Manuscript received 25 April 2003; revised 7 April 2004
Communicated by Ladislav Hluchý

Abstract. This paper presents a market mechanism for dynamic resource allocation in computational grid. Grid market is described that consists of two economic agent types; it allows agents representing various grid resources to coordinate their resource allocation decisions without assuming a priori cooperation. The grid task agents buy resources to complete tasks. Grid resource agents charge the task agents for the amount of resource capacity allocated. Grid resource allocation problem is presented as grid user utility optimization. Given grid resource agent's pricing policy, the task agent optimization problem is to complete its job as quickly as possible when spending the least possible amount of money. This paper provides a resource allocation and pricing algorithm. Experiments are made to compare the performance of the price-directed resource allocation with conventional Round-Robin allocation.

Keywords: Grid, resource management, market, agent

1 INTRODUCTION

Grid computing is an emerging technology that promises to unify resources and computing power in many organizations. It is widely used to solve large-scale problems in engineering and science area. Since the Grid computing technology is still in a very early stage, there are still a few works on the building of efficient resources manager on the Grid [1–4]. Carsten Ernemann [8] addresses the idea of applying economic models to the scheduling task. In [8] a scheduling infrastructure and a market-economic method is presented. The efficiency of this approach in terms of response and wait time minimization as well as utilization is evaluated by simulations with real workload traces. The evaluations show that the presented economic scheduling algorithm provides similar or even better average weighted response-times as common algorithms like backfilling. This is especially promising as the presented economic models have additional advantages as e.g. support for different price models, optimization objectives, access policies or quality of service demands. Grid Architecture for Computational Economy (GRACE) [9, 21] proposes a distributed computational economy as an effective metaphor for the management of resources and application scheduling. It proposes an architectural framework that supports resource trading and quality of services based scheduling. It enables the regulation of supply and demand for resources; provides an incentive for resource owners to participate in the Grid; and motivates the users to trade-off between deadline, budget, and the required level of quality-of-service. It also demonstrates the capability of economic-based systems for peer-to-peer distributed computing by developing users' quality-of-service requirements driven scheduling strategies, algorithms, and systems. Chun and Culler [22] propose CPU timeshare allocations are governed by a market economy that optimizes user value. Their approach does not address the issue of market equilibrium; rather, as with auctions, it sets prices locally, but prices are set based on priority. Nimrod-G [10, 17] is a Grid resource broker that allows managing and steering task farming applications on computational Grids. It uses an economic model for resource management and scheduling. Users formulate parameter studies using a declarative parametric modeling language or GUI with the experiment being run on the Grid. Nimrod-G provides resource discovery, resource trading, scheduling, resource staging on Grid nodes, result gathering, and final presentation to the user.

Compared with other grid management, our model proposes an economic solution to the problem of heterogeneous demand in the grid. The desire of end user is represented by utility functions to allow them to specify resource requirements and preference parameters. This method is targeted to solve heterogeneous demand in the grid. It is a user-centric scheme in which scheduling decision is directed by the users requirements. Experiments are made to compare the performance of the price-directed resource allocation with conventional Round-Robin allocation.

The rest of the paper is organized as follows. Section 2 describes grid market for grid resource management; Section 3 presents the grid resource allocation problem; Section 4 describes allocation and pricing problem; Section 5 describes initial experiment; Section 6 concludes the paper.

2 GRID MARKET FOR GRID RESOURCE MANAGEMENT

The grid market consists of two economic agent types (see Figure 1): the grid resource agents that represent the economic interests of the underlying resources of the computational grid, the grid task agents that represent the interests of grid user using the grid to achieve goals. Grid market has information about the‘ locations of current resource providers in the grid and about their prices. Whenever a grid resource agent in the grid decides to sell its resources, change its pricing structure, or update available capacity, it will spawn an agent to find grid markets and update the advertised information. The grid market then provides this information to other agents wishing to know about resource providers. Whenever a new grid task agent is created, it is first given an endowment of electronic cash to spend to complete its task. If that agent either refuses to make a purchase under that level of availability or that price structure, or if the task agent does not purchase all of the available capacity, the resource agent offers the remaining capacity to the next task agent. We assume that when a task agent purchases a portion of the resource, it is guaranteed that the task agent continues to receive resource uninterrupted from the resource agent until its task is completed. The price that the agent pays, per second of resource capacity, is the same for as long as he continues to use the purchased rate. The agent makes no guarantee to the resource provider and may leave the queue or leave the processor at any time. The user makes this decision by keeping up-to-date on the resources and prices offered by other resource providers on the grid. This can be done by periodically spawning agents that travel to grid markets and return with price and resource quotes [5, 6, 14–16].

A grid resource agent is used at the source node in the grid and is deployed at the entry node. The Grid resource agents have varied computational resource capacity, and the computational resource capacity is shared among the grid task agents. The grid resource agents charge the task agents for the portion of the computational resource capacity occupied. We assume that the grid resource agents of a grid do not cooperate, probably due to high messaging and processing overheads associated with cooperative allocating. Instead, they act non-cooperatively with the objective of maximizing their individual profits. The grid resource agents compete among each other to serve the task agents. The task agents do not collaborate either, and try to purchase as much computational resource as possible with the objective of maximizing their net benefit. The agents communicate by means of a simple set of signals that encapsulate offers, bids, commitments, and payments for resources. We couple the resources and payments with the offers and requests, respectively. This reduces the number of steps involved in a transaction (committing agents to their payments and offers ahead of the market outcome), and so increases the speed of the system's decision making. To enforce these rules the interactions between the two agent types are mediated by means of market mechanisms. In our market mechanisms, agent communication is restricted to setting a price on a single unit of a known grid resource. Therefore, agents set their prices solely on the basis of their implicit perception of supply and demand of grid resource at a given time. When a resource is scarce, grid task agents have to increase the prices they are willing to buy, just as resource agents decrease the price at which they are willing to offer the resource. In our model, agents perceive supply and demand in the market through price-directed market-based algorithm that will be described in Section 4.

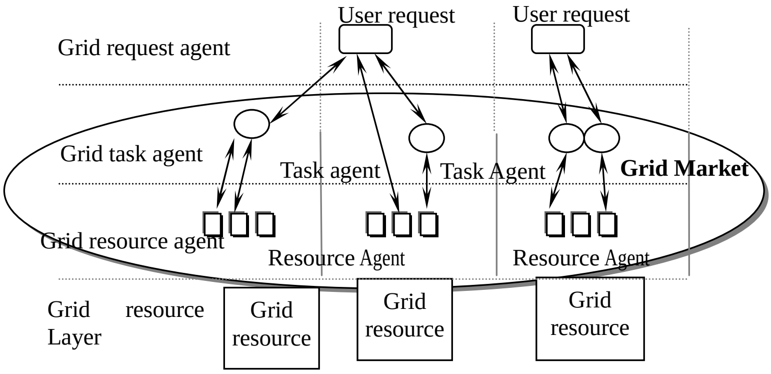

Fig. 1. Grid market model

Grid resource agents publish resource/service descriptions to Grid Market Manager. A grid resource agent can appear or leave the grid at any time. Resource grid agent provides services that can be used by other grid agents. Grid resource agents can have multiple accessing interfaces, though an agent does not need to publish all of its interfaces for use. Each published interface is advertised as an agent entry point. An agent entry point is a URI that uniquely identifies an agent's interface. Grid resource agent entry points are used by other grid agents to establish direct point-to-point connection between two grid agents [18].

A grid resource agent has a set of n resources units of grid resource' computation capacity required to serve task agent's requirement, that it can sell under market control. At time t, the price to be asked for each of these units is stored in a vector $p_t = \{p_t^1, p_t^2, \ldots, p_t^n\}$ with the range of possible prices being zero to infinity, $p_t \in (0, \infty)$, for each member of the vector $i = 1, \ldots, n$ and each time period t. At time $t = 0$ the prices for each unit are randomly (uniformly) distributed on $[0, H]$ where H is the initial upper limit on prices asked. Given its baseline level, the resource agent attempts to maximize its income. When x units have been allocated, the remaining $n - x$ units are offered for sale simultaneously [7]. Suppose that of the $n - x$ units offered for sale in a given period t, the m units with the lowest prices are successfully sold. The prices in the vector are updated by using price-directed allocation algorithm. Thus the grid resource agent increases or decreases the price of any unit by a small amount after each negotiation. Here is obtained from a uniform random distribution. This approach was chosen so that the prices of each resource should adapt to the dynamic demand on the grid.

Grid resource agents sell the underlying resources of the grid. A grid task agent that represents the grid user makes buying decisions within budget constraints to acquire computation resources. The offers placed on the grid market by grid resource agents are allocated to grid task agent. Grid task agents buy computation resources solely on the basis of the most recent price information they have. Grid task agents initiate competing for grid resource by signaling that they wish to buy resources to complete certain tasks. The task agent retains a vector of prices that it is willing to pay for resources. The task agent tries to maintain its resources at a level optimal units denoted by $x_i^{j^*}$ discovered through gradient climbing adaptation to the behavior of the market described in Section 4. The most profitable value of $x_i^{j^*}$ is obtained by adjusting it according to changing profit during ongoing buying and selling episodes. The price paid for each resource agent should be as low as possible without failing to obtain the resource. Therefore the task agent makes a request for each resource that it needs separately. If a request was rejected, the agent increases the price it will send to resource agent at the next negotiation. If a request was accepted, the agent reduces the price it pays for that resource in subsequent negotiations.

Grid task agents and grid resource agents do not communicate directly with one another or amongst themselves. All interactions are by the means of grid market. The grid market also broadcast the prices at which trades are agreed, so the agents have more information upon which to base their trading behavior. The negotiation between agents is mediated by means of a grid market. It allows multiple grid task agents and grid resource agents to negotiate simultaneously, it provides a dense set of market price information and it allows supply and demand to be reconciled at the same time. Grid market provides a means to complete institutionally mediated bargaining in one shot that would take an indeterminate time using iterated market allocation algorithms. The resources exchanged at the grid markets are the right to use slices of computation resource of grid resource agent, which when taken together, provide the necessary capacity to grid task agent. The grid markets use price-directed allocation algorithm that will be described in Section 4. In this algorithm an initial set of prices is announced to the task agent. In each iteration, grid resource tasks individually determine their optimal allocation and communicate their results to the grid resource agent. Grid resource agents then update their prices and communicate the new prices to the task agents and the cycle repeats. Prices are then iteratively changed to accommodate the demands for resources until the total demand equals to the total amount of resources available. The task agent's utility maximization is also considered.

3 GRID RESOURCE ALLOCATION PROBLEM

The grid task agents buy resources to complete tasks. Grid resource agents charge the task agents for the amount of resource capacity allocated. However, there are multiple grid task agents competing to buy the grid resource agent's computation resource. We investigate the effect of this competition on the system model. Specifically, we show that such a price competition leads to the optimal grid resource allocation strategy for the grid task agents. This approach provides a dynamical and distributed algorithm for determining the resource allocation in the grid. The basic problem of grid market is to divide some resources among a number of grid task agents via a sound market mechanism. In order for the mechanism to be economically efficient, the utility of grid task agents can be maximized when agent's requirements are rational. Each grid task agent has a utility function telling how much it is worth for the agent to have a certain amount of the resources. The problem is formulated as a resource allocation problem that should be solved via a market mechanism. In this section we find the task agent's optimal allocating strategy under a certain grid resource agents pricing scheme. First, we formulate the utility function of grid task agent, and then give optimization solution to grid resource allocation.

3.1 Grid Agent Description

A market mechanism's efficiency depends on the consumers' ability to assess their needs and then make rational decisions that maximize their utility. Here, utility is a measure of the pleasure a market participant derives from consumption of a good. We are concerned with two attributes that affect grid task agents. The first factor is cost. We limit every task agent's monetary resources, so a task agent that pays a higher price for grid resource effectively limits the amount of utility it can generate in the future. Currency can be considered an abstract good representing future consumption. The second factor is completion time. This depends on resource congestion and the hardware providing service. There could be other qualities besides completion time. In information retrieval tasks, accuracy is another factor.

Grid task agents want to complete a set of jobs in a given sequence by purchasing resources from grid resource agents located throughout the grid. An agent begins with an endowment of E_i to spend to complete its task and wishes to minimize the total time taken to complete a sequence of jobs given its budget constraint. We assume that there are K types of resources and that each agent may needs many types to complete a job. Assume that there is a set $K = \{1, 2, \ldots, K\}$ of different types of resources that the grid allocates at each grid task agent in order to complete the task. For example, if storage systems and databases are the two types of resources that the grid allocates in order to complete the task, then $K = \{1, 2\}$. In this case, $k = 1$ refers to storage systems and $k = 2$ refers to databases. The agent's task can be represented as the sequence $(q_{ij})_{j=1}^{j=K}$, where q_{ij} is the size of i^{th} task agent' s j^{th} job.

Let u_i^j be the price paid to j^{th} resource agent per time unit by the i^{th} task agent. Let be the total investment of the i^{th} task agent, which is defined in (4.1). N grid task agents compete for grid resources with finite capacity. The resource is allocated using a market mechanism, where the partitions depend on the relative payments sent by the grid task agents. We assume that each task agent submits u_i^j to the grid resource agent. Then, $u^j = [u_1^j \ldots u_N^j]$ represents all payments of grid task agents for j^{th} resource agent. Let p_j denote the price of the unit computational resource in resource agent j. Let the pricing policy, $p = (p_1, p_2, \ldots, p_j)$, denote the set of unit computational resource prices of all the resource agents in the grid.

$$u_i = \sum_j u_i^j \tag{1}$$

Let x_i^j be the fraction of resource units allocated to task agent i by resource agent j. If i^{th} task agent's payment in the j^{th} resource agent is u_i^j, then the total computation resource units allocated to task agent i is

$$x_i^j = \frac{u_i^j}{p_j}. \tag{2}$$

The i^{th} agent receives resources proportional to its payment relative to the sum of the resource agent's revenue, c_j is the capacity in computational units of j^{th} grid resource agent. u_i^j is the amount that the i^{th} agent pays for resource j, r_i^j is the capacity that i-th agent pays receives.

$$r_i^j = x_j \frac{u_i^j}{p_j}$$

The time taken by the i^{th} agent to complete its job is: $t_i^j = \frac{q_{ij}}{c_j x_i^j}$ and the expenses are: $M_i^j = u_i^j \frac{q_{ij}}{c_j x_i^j}$.

The goal of each task agent is to complete its job as quickly as possible when spending the least possible amounts of money. q_{ij} is the size of i^{th} task agent's j^{th} job. Since the grid user wishes to minimize both the time, $\sum_{j=1}^{N} \frac{q_{ij}}{c_j x_i^j} + D$, and money $\sum_j u_l^j$ it spends. The utility function $U(x_i^j)$ of the grid task agent is defined as (3).

$$U(x_i^j) = -K \left(\sum_{j=1}^{N} \frac{q_{ij}}{c_j x_i^j} + D \right) - \sum_j u_i^j \tag{3}$$

Where D is the delay, which includes waiting times, transfer times between various nodes in the grid. K is the relative importance of costs and times to complete grid task, an agent with larger value of K would indicate a greater preference to reduce its completion time. When K is 1, costs and times are equally important.

3.2 Grid User Benefit Optimization Under Budget Constraint

Every grid task agent tries to maximize itself benefit regardless of others subject to the availability of budgets and complete time limits. For a given grid resource pricing policy P, the task agent optimization problem (S) can be written as (4).

$$\text{(S)} \qquad \max U(x_i^j) \tag{4}$$

$$\text{s.t. } E_i \geq \sum_j x_i^j p_j$$

Constraint is a budget constraint, which says that the aggregate sum of all costs of each task agent cannot exceed its total budget. E_i is endowment given to an agent. Our objective is to choose optimal x_i^j.

$$\sum_{i=1}^{N} x_i^j = 1 \tag{5}$$

Indicates a grid resource is divisible, that can be shared among many grid task agents.

We substitute $x_i^j = \frac{u_i^j}{p_j}$ into $U(x_i^j)$ to obtain (6):

$$U(x_i^j) = -K\left(\sum_{j=1}^{N} \frac{q_{ij}}{c_j x_i^j} + D\right) - \sum_j x_i^j p_j. \tag{6}$$

We compute the optimum by deriving the derivative of $U(x_i^j)$ with respect to x_i^j as (7).

$$U'(x_i^j) = \frac{\mathrm{d}U(x_i^j)}{\mathrm{d}x_i^j} = \sum_{j=1}^{N} \frac{q_{ij}}{c_j\left(x_i^j\right)^2} - p_j \tag{7}$$

Then, the second derivative of $U(x_i^j)$ with respect to x_i^j is (8).

$$U''(x_i^j) = \frac{\mathrm{d}^2U(x_i^j)}{\mathrm{d}\left(x_i^j\right)^2} = -K\sum_{j=1}^{N} \frac{q_{ij}}{c_j\left(x_i^j\right)^3} \tag{8}$$

$U''(x_i^j) < 0$ is negative due to $0 < x_i^j < 1$. The extreme point is the unique value maximizing the agent's utility and is optimal resource demand for grid resource agent. Grid task agent's utility is a convex function of x_i^j. A common method of optimizing convex function is to apply Lagrangian. The Lagrangian for the task agent's utility is $L(x)$ (9).

$$L(x_i^j) = -K\left(\sum_{j=1}^{N} \frac{q_{ij}}{c_j x_i^j} + D\right) - \sum_j x_i^j p_j - \lambda\left(\sum_j x_i^j p_j\right) \tag{9}$$

where λ is the Lagrangian constant. From Karush-Kuhn-Tucker Theorem we know that the optimal solution is given $\partial L(x)/\partial x = 0$ for $\lambda > 0$.

$$\frac{\partial L(x_i^j)}{\partial x_i^j} = K\sum_{j=1}^{N} \frac{q_{ij}}{c_j\left(x_i^j\right)^2} - (1+\lambda)p_j \tag{10}$$

Let $\partial L(x_i^j)/\partial x_i^j = 0$ to obtain (11):

$$x_i^j = \left(\frac{Kq_{ij}}{(1+\lambda)p_j c_j}\right)^{\frac{1}{2}}. \tag{11}$$

Using this result in the constraint equation, we can determine $\theta = \lambda + 1$ as

$$\theta^{-\frac{1}{2}} = \frac{E_i}{\sum_{k=1}^{N} p_k \left(\frac{Kq_{ik}}{c_k p_k}\right)^{\frac{1}{2}}}. \tag{12}$$

We substitute (12) into (11) to obtain (13):

$$x_i^{j^*} = \frac{\left(\frac{q_{ij}}{p_j c_j}\right)^{\frac{1}{2}} E_i}{\sum_{k=1}^{N} p_k \left(\frac{q_{ik}}{c_k p_k}\right)^{\frac{1}{2}}}. \tag{13}$$

$x_i^{j^*}$ is the unique optimal solution to the optimization problem (S).

4 ALLOCATION AND PRICING ALGORITHM

An iterative resource allocation and pricing algorithm is designed to solve the grid task agent resource allocation problem. In each iteration, the grid user individually solves its fees to pay, adjusts its grid resource demand and notifies the grid about this change. After the new grid resource demand is observed by the grid resource agent, it updates its price accordingly and communicates the new prices to the grid task agent, and the cycle repeats. To illustrate how grid task agent adjusts its fees to pay, we define the demand function $D(p) = R \to R$, which is defined as the quantity of resources that the agent would desire if the price is p. $D(p)$ can be obtained by optimal solution $u_i^{j^*}$ to S (4) problem.

$$D(p) = \frac{u_i^{j^*}}{p_j} \tag{14}$$

The iterative algorithm that computes the price and resource allocation is then given as follows.

4.1 Algorithm 1 Grid Resource Unit Price Calculation and Resource Allocation

Grid resource agent part algorithm at iteration n

(1) receives grid resource demand $x_j(n)$ from grid task agents;

(2) computes a new price according to the following formula

$$p_j^{(n+1)} = \max\left\{\varepsilon, p_j^{(n)} + \eta\left(x^j(n)p^{(n)} - c_j\right)\right\} \tag{15}$$

where $x^j = \sum_i x_i^j$, $\eta > 0$ is a small step size parameter, n is iteration. Let $\varepsilon > 0$ be a sufficiently small constant preventing prices to approach zero. This algorithm is consistent with the law of supply and demand: if the demand for grid resource exceeds the capacity supply c_j, then the price $p_j^{(n+1)}$ is raised; otherwise, the price is reduced;

(3) communicates new price $p_j^{(n+1)}$ to all grid task agents.

Grid task agent part algorithm at iteration n

(1) receives from the grid resource agent j the price p_j which is calculated by (15)

(2) calculates its optimal resource demand according to $u_i^{j^*}$ to maximize $U(u_i^j)$

$$x^j(n+1) = \frac{u_i^{j^*(n)}}{p_j^{(n)}} \tag{16}$$

(3) communicates new grid resource demand $x^j(n+1)$ to grid resource agents.

5 EXPERIMENTS

The goal of this experiment is to compare the performance of a decentralized economic approach based on the price-directed resource allocation algorithm with conventional Round-Robin allocation algorithm. To do this, both approaches are evaluated experimentally by means of simulations. In the Round-Robin allocation scheme, no pricing is used. The incoming task queries are matched with the next available resource offer, which meets the task's constraints but which is usually not the best. First we introduce the configuration of simulation, then give the experiment design and results.

The simulator was developed to test the price-directed allocation algorithm. It is implemented on top of the JAVASIM network simulator. Different agent types can be instantiated, namely grid client, grid task agents, and grid resource agents. Grid resources to be allocated encompass computation service access, bandwidth and storage. The simulation is built on a TCP/IP network model supported by JAVASIM. The physical network topology is specified in the input of the simulator. The experiment is to study characteristics of price-directed allocation algorithm with Round-Robin algorithm in terms of response time and resource allocation efficiency. Grid systems are randomized in various sizes: 100, 500, 1000, and 2000 nodes. In the experiments we change some of test parameters, such as the size of grid that is denoted by S in the following figures, resource requests intensity is denoted by I. We devise requests intensity with 50 ms, 100 ms, 200 ms and 400 ms.The experiment is to randomly submit 250 grid requests and schedule them to the specific grid resource based on price-directed resource allocation and Round-Robin allocation. Arrival time of each resource request is determined randomly in exponential distribution with the mean of 200 ms, but we will change the values of arrival time when testing effect of requests intensity on response time and resource allocation efficiency. All nodes are initially no loads. During the time of experiment, grid resource requests are generated by the grid user agent. After this initial period, the number of tasks that is statistically expected to be generated during an interval of 100 time units is considered in the result. To allow grid task agents to complete tasks, an additional margin of 300 time units is provided. For the simulated scenarios it does not matter how many grid user agents there are in the system. Therefore, we use only one grid user agent for the generation of grid resource agent. The size of data carried by the grid task agent is denoted by D; all tasks have the same computation size $D = 25\,\mathrm{KB}$. Task deadlines are not used. The default value of the task price denoted by P is 100. There are 25 grid resource agents in the system. All gird resource agents have the same resource size denoted by R, set $R = 100$. All parameters are summarized in Table 1. Each measurement is run 30 times with different seeds. These experimental configurations are to bring up performance of resource allocation algorithm as many as possible. Interesting variables are recorded and plot average results in Figure 2 and Figure 3 for response times and resource allocation efficiency, respectively.

Parameter	Value
Number of nodes in a cluster	exp(25)
Number of nodes	100, 500, 1000, 2000
Reschedule Interval	600 seconds
Number of Jobs	250
Requests intensity (ms)	50, 100, 200, 400
Arrival time (ms)	200

Table 1. Simulation parameters

Firstly, we have measured the response times of price-directed allocation and Round-Robin allocation when using the following parameters for the test: ($I = 200\,\mathrm{ms}$). Response time measures the time observed by the grid client to access the requested grid resources. It is influenced by the size of the grid, the available connections and bandwidth, and especially by the necessary mechanisms to establish a working link between grid task agent and grid resource agent. For comparing different size grid, a lower average response time is considered to be better. The results are shown in Figure 2. From the results in Figure 2, for Round-Robin allocation, the response time value seems to depend on the grid size. Price-directed allocation and Round-Robin allocation present the good results for this small size grid. However, when the size of grid is larger, Round-Robin allocation is decreasing quickly; the response time using price-directed allocation can be as much as 44 % shorter than that using the Round-Robin allocation. On big grid, Round-Robin allocation takes more time to allocate appropriate resources. As shown in Figure 2, for different size grid, the price-directed allocation outperforms the conventional Round-Robin allocation.

Secondly, we measured the resource allocation efficiency of price-directed allocation and Round-Robin allocation when using the following parameters for the test:

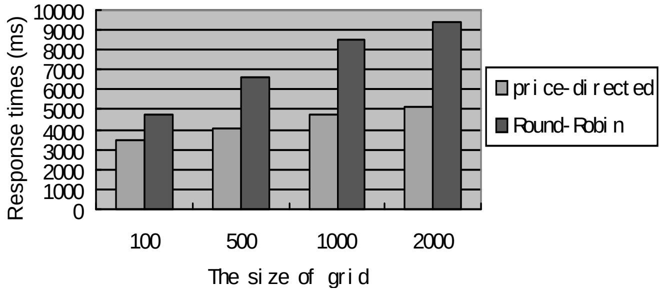

Fig. 2. Comparison of response time

($I = 200\,\mathrm{ms}$). Resource allocation efficiency indicates the ratio of grid resource requests, for which the grid resource agent grants to provide a resource, to all sent grid resource requests. In other words, it measures how many requests a grid client has to send until a resource agent accepts its demand and grants access. As the request messages waste up bandwidth, higher resource allocation efficiency is deemed to be better both for the individual grid client agent and for the whole grid as a whole. The results are shown in Figure 3. It becomes clear that both allocation schemes work best under small size grid. The Round-Robin allocation achieves to match nearly 98 % of all requests in small size grid scenario, with price-directed allocation closely behind. However, as grid size increases, the Round-Robin allocation soon loses comparably more performance than the price-directed allocation. Under large size grid, the decrease of the results for Round-Robin allocation is lower than in the small size. Resource allocation efficiency using price-directed allocation is as much as 27 % larger than that using the Round-Robin allocation. With varying grid size, the result decreases for both methods similarly.

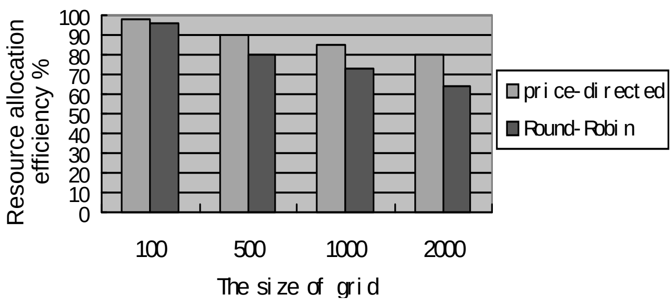

Fig. 3. Resource Allocation Efficiency

From the above performance comparisons, some conclusions can be drawn. In most of the test cases, the price-directed allocation is more efficient than the Round-Robin allocation to allocate grid resource in test application. When grid size is increasing, there are more merits to use the price-directed allocation to schedule grid resource; the price-directed allocation has better performance than usual Round-Robin allocation.

6 CONCLUSIONS

This paper presents a market-based approach to computational grid resource management. A realistic model for the relationship between the grid task agent and grid resource agent is presented. The grid task agents buy resources to complete tasks. Grid resource agents charge the task agents for the amount of resource capacity allocated. However, there are multiple grid task agents competing to buy the grid resource agent's computation resource. Given grid resource agent's pricing policy, the task agent optimization problem is provided. This paper provides a price-directed market-based algorithm for solving the grid task agent resource allocation problem. The results of experiment show the price-directed allocation has better performance than usual Round-Robin allocation.

Acknowledgements

The work is supported by National Natural Science Foundation of China and NSF of Hubei Province.

REFERENCES

[1] Foster, I.—Kesselman, C.: The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, 1999.

[2] Krauter, K.—Buyya, R.—Maheswaran, M.: A Taxonomy and Survey of Grid Resource Management Systems, Software: Practice and Experience. Vol. 32, 2002, No. 2, pp. 135–164.

[3] Lalis, S.—Karipidis, A.: An Open Market-Based Framework for Distributed Computing over the Internet. Proceedings of the First IEEE/ACM International Workshop on Grid Computing (GRID 2000), Dec. 17, 2000, Bangalore, India, Springer Verlag Press, 2000, pp. 36–46.

[4] Wolski, R.—Plank, J.—Brevik, J.—Bryan, T.: Analyzing Market-Based Resource Allocation Strategies for the Computational Grid. International Journal of High-performance Computing Applications, Sage Publications, Vol. 15, 2001, No. 3, pp. 258–281.

[5] Bredin, J.—Kotz, D.—Rus, D.: Utility Driven Mobile-Agent Scheduling. Technical Report PCS-TR98-331, Dartmouth College, 1998.

[6] Chunlin, L.—Layuan, L.: An Agent-Oriented and Service-Oriented Environment for Deploying Dynamic Distributed Systems. Journal Computer Standard and Interface, Elsevier, Vol. 24, 2002, No. 4, pp. 321–334.

[7] Ferguson, D. F.—Yemini, Y.—Nikolaou, Ch.: Microeconomic Algorithms for Load Balancing in Distributed Computer Systems. Proceedings of the 8th Interna-

tional Conference on Distributed Computing Systems, San Jose, IEEE-CS Press, 1988, pp. 491–499.

[8] Ernemann, C.: Economic Scheduling in Grid Computing. In Proc of 8th International Workshop Job Scheduling Strategies for Parallel Processing, Lecture Notes in Computer Science, UK, July 24, 2002, pp. 128–152.

[9] Buyya, R.—Stockinger, H.—Giddy, J.—Abramson, D.: Economic Models for Management of Resources in Peer-to-Peer and Grid Computing. In Proceedings of International Conference on Commercial Applications for High-Performance Computing, SPIE Press, 2001, pp. 13–25.

[10] Abramson, D.—Buyya, R.—Giddy, J.: A Computational Economy for Grid Computing and Its Implementation in the Nimrod-G Resource Broker. Future Generation Computer Systems Journal, Vol. 18, 2002, No. 8, pp. 1061–1074.

[11] Varian, H. R.: Microeconomic Analysis (Third Ed.). W. W. Norton & Company Inc., 1992.

[12] Wellman, M.: Market Oriented Programming: Some Early Lessons, in Market-Based Control a Paradigm for Distributed Resource Allocation. Ed. S. H. Clearwater, World Scientific Press, 1996.

[13] Jennings, N. R.—Wooldridge, M. R.: Agent Technology Foundations, Applications and Markets. Springer Verlag, 1998.

[14] Chunlin, L.—Zhengding, L.—Layuan, L.—Shuzhi, Z.: A Mobile Agent Platform Based On Tuple Space Coordination. Journal of advances in engineering software, Elsevier, Vol. 33, 2002, No. 4, pp. 215–225.

[15] Chunlin, L.—Layuan, L.: Integrate Software Agents And CORBA In Computational Grid. Journal of Computer Standards and Interfaces, Elsevier, Vol. 25, 2003, No. 4, pp. 357–371.

[16] Chunlin, L.—Zhengding, L.—Layuan, L.: Design and Implementation of a Distributed Computing Environment Model for Object-Oriented Networks Programming. Journal of Computer Communications, Elsevier, Vol. 25, 2002, No. 5, pp. 517–522.

[17] Frey, J.—Tannenbaum, T.—Livny, M. et. al: Condor-G: A Computation Management Agent for Multi-Institutional Grids. Cluster Computing, Vol. 5, 2002, No. 3, pp. 237–246.

[18] Chunlin, L.—Layuan, L.: Combine Concept of Agent and Service to Build Distributed Object-Oriented System, Future Generation Computer Systems, Elsevier, Vol. 19, 2003, No. 2, pp. 161–171.

[19] Buyya, R.—Abramson, D.—Giddy, J.: Nimrod/G: An Architecture for a Resource Management and Scheduling System in a Global Computational Grid. International Conference on High Performance Computing in Asia-Pacific Region (HPC Asia 2000), Beijing, China. IEEE Computer Society Press, USA, 2000.

[20] Heiser, G.—Lam, F.—Russell, S.: Resource Management in the Mungi Single-Address-Space Operating System, Proceedings of Australasian Computer Science Conference, February 4–6, 1998, Perth Australia, Springer-Verlag, Singapore, 1998.

[21] Buyya, R.—Giddy, J.—Abramson, D.: A Case for Economy Grid Architecture for Service-Oriented Grid Computing. 10$^{\text{th}}$ IEEE International Heterogeneous Com-

puting Workshop (HCW 2001), in conjunction with IPDPS 2001, San Francisco, California, USA, April 2001.

[22] Chun, B. N.—Culler, D. E.: Market-Based Proportional Resource Sharing for Clusters. Millenium Project Research Report, Sep. 1999.

Li Chunlin was born in 1974. She received the M.E. in computer science from Wuhan Transportation University in 2000, and Ph.D. degrees in computer software and theory from Huazhong University of Science & Technology in 2003. Currently she is a lecturer of computer science in Wuhan University of Technology. Her research interests include mobile agent, distributed computing and computational grid. She has published over 15 papers in international journals.

Li Layuan was born in 1946. He received the BE degree in communication engineering from Harbin Institute of Military Engineering, China in 1970 and the ME degree in communication and electrical systems from Huazhong University of Science and Technology, China in 1982. He academically visited Massachusetts Institute of Technology, USA in 1985 and 1999. Since 1982, he has been with the Wuhan University of Technology, China, where he is currently a Professor and Ph.D. tutor of computer science, and Editor in Chief of the Journal of WUT. He is Director of International Society of High-Technology and paper reviewer of IEEE INFOCOM, ICCC and ISRSDC. He was the head of the Technical Group of Shaanxi Lonan PO Box 72, Ministry of Electrical Industry, China from 1970–78. His research interests include high speed computer networks, protocol engineering and image processing. Professor Li has published over one hundred and fifty technical papers and is the author of six books. He also was awarded the National Special Prize by the Chinese Government in 1993.