The Optimal Choice by Resource Consumers in Grid Market

Laiping Zhao, Mingchu Li, Weifeng Sun,
Dalian University of Technology
Lab. of Infor. Security & Grid Tech.
Dalian, China
zhaolaiping@gmail.com,
li_mingchu@yahoo.com
wfsun@dlut.edu.cn

Kouichi SAKURAI and Yizhi Ren
Kyushu University
Dept. of C.S. & Comm. Engineering
Fukuoka, Japan
sakurai@csce.kyushu-u.ac.jp,
ren@itslab.csce.kyushu-u.ac.jp

Abstract

Market mechanism is effective to encourage more people to participate in the Grid environment. Applying economic principles to Grid is deemed promising to improve the overall value provided by such participants. However, resource scheduling algorithms are different depending on different needs. In this paper, we mainly focus on the consumers' own interest. Assuming that the consumers are selfish and what they concern about is lower cost and higher success rate, we improve the HRED algorithm using hierarchical mechanism and resource selection mechanism, which cooperate with resource providers' trust degree, making it more suitable, practical and credible for Grid market. The experiments prove that the improved HRED algorithm works faster especially when there are large numbers of resource providers in Grid market, and it also gives much higher success rates in the non-credible Grid environment.

1. Introduction

1.1. Background

In Grid environment, the geographically distributed heterogeneous resources are shared and aggregated to provide a variety of services for users transparently. Economics based Grid computing provides an incentive mechanism for resource providers, who can get corresponding profits from the sharing of their resources. Once the Grid gets popular in our life, while some companies, organizations or individuals would like to submit their jobs to the Grid, wishing to get the resources and services they want, some organizations or individuals also would like to earn "money" from contributing their resources to the Grid.

In that case, a market mechanism is necessary for the Grid, and GRACE [2], Nimod/G [3], Tycoon [10], and Bellagio [1] are all good examples, which have done a lot of work on both the research and deployment.

1.2. Motivation

Recently, there has been an increased research focusing on applying Austrian economist F.A. von Hayek's notion of 'Catallaxy' to market-driven Grid computing [7]. From the idea of 'Catallactics', each participant tries to maximize their own gain based on the information available to them. Participants are all self-organizing, follow their own interest, and maximize their own utility.

Taking the execution time, economic cost and trust degree as illustrations, consumers may want to get resources or services from the Grid at a lower cost, in a shorter time or at a higher success rate [4]. A scheduling policy is needed to meet these demands and maximize their utilities as great as possible.

1.3. Previous work

Considering the time and cost factors, Rajkumar Buyya [2] proposes the cost optimization scheduling algorithm, time optimization algorithm and conservative time optimization algorithm, and the DBC scheduling algorithm is also given on this basis trying to solve cost-time optimization problem [4], which selects the resource with the lowest cost for which the expected response time is within the deadline of the job, and the cost of processing the job does not exceed the budget of the owner of the job. Subodha Kumar et al. formulate the consumers' cost minimization problem, and propose the HRED algorithm [9] which gives an near-optimal solution on minimizing economic cost scheduling. However, these algorithms do not consider their practicality in a large-scale Grid environment. When there are large numbers of resource providers in the

978-0-7695-3737-5/09 $25.00 © 2009 IEEE
DOI 10.1109/UIC-ATC.2009.75

Grid market, the scheduling efficiency is very important, but these algorithms are not suitable if many participants exist.

FirstPrice [8] sorts and greedily schedules jobs based on a schedule that will maximize per unit return. FirstReward [8] considers both the per unit return for accepting a task and the risk of losing gains in the future. It utilizes a Present Value approach that considers the utility countered by a discount rate on future gains that could be made. However, they doesn't reactive to the malicious behavior of some providers in the Grid market.

In addition to traditional algorithms, heuristic algorithms such as genetic algorithm and ant colony algorithm are also introduced into resource scheduling. Paper [12] and [14] make use of genetic algorithm to assign jobs to resources: each job is assigned to one resource, and the resources serial numbers are encoded as genes, after crossover and mutation operations, new chromosomes that are indexed with the site assignment for jobs will be generated, then sift out the better results with a fitness function and repeat the whole process a number of times until the best solution is found.

Applying ant colony algorithm into resource scheduling is introduced in [5]. Each ant stands for a job, and the arcs correspond to resources. A set of artificial ants is initially created. Each ant starts with unscheduled job in machine and then they build a job tour in machines until a feasible solution is constructed. In the scheduling process, local update and global update are behaved to get the newest pheromone for the next job submission.

Compared with traditional algorithms, these heuristic algorithms are always inefficient in resource scheduling rate. Moreover, because of the large number of resources, the encoding method in genetic algorithm and the pheromone indicator matrix in ant colony algorithm are too large to maintain.

1.4. Challenging issues

T. Eymann et al. have investigated the issues and requirements of implementing an electronic Grid market based on the concept of "Catallaxy" [7], and found that the multi-attribute resource allocation is an NP-complete problem and also non-approximable. The HRED algorithm [9] is proposed to address this problem, which provides a near-optimal solution for the cost minimization problem and can be extended to time minimization problem easily.

However, both the reliability of scheduled resources and the efficiency of scheduling algorithm are as important as the scheduling algorithm itself. In a Grid market, the number of the resource providers is large while maybe they are also malicious. Therefore, the HRED algorithm should be improved to take these issues into consideration, and be designed practically for the large-scale distributed environment.

1.5. Our contribution with comparing to related works

Based on the challenging issues, our work mainly focuses on three parts:

- Apply hierarchical scheduler and resource selection mechanism in HRED algorithm, improving the scheduling efficiency.
- Quantify the risk to part of cost, so that cost minimization scheduling results are also credible.
- Evaluate the efficiency and success rates through simulating large numbers of resource providers.

Compared with original HRED, our solution can schedule much faster especially when there are large numbers of resource providers in the market grid,

$$Time_{I-HRED} = \frac{n}{N} Time_{HRED}, \quad (1)$$

where N is the total number of resource providers and n is the size of set RP_i (Section 3.2).

Meanwhile, with the trustworthy resource providers, the task success rates are much higher than original HRED. The simulation experiment shows that the success rates are always above 0.8 than original 0.5 (Section 4).

2. The Grid market

The Grid market is constructed by resource users, schedulers and providers (Fig. 1), where users submit their jobs to the Grid, schedulers accept request and search for appropriate resource providers, and resource providers complete the jobs. Every resource owner can be one of resource providers, and every user can be a resource consumer.

Economics based Grid environment mainly aims at two points: encouraging more people to participate in the Grid and optimizing global resource allocation. Fortunately, both user's utility and the global utility can be optimized simultaneously under the perfect competitive market equilibrium [11]. We assume that the Grid environment is a perfect competitive market, and users arrange their purchasing behavior rationally based on the resource price state. According to economic principles, the market will get into a desirable equilibrium state finally. We are intent on researching resource consumers' choice here.

The scheduling process is defined like below:

1. User submits his jobs to the local scheduler (the Scheduler1 in Fig. 1).
2. Local scheduler receives jobs and uses scheduling algorithm to search for appropriate resource providers from local domain.

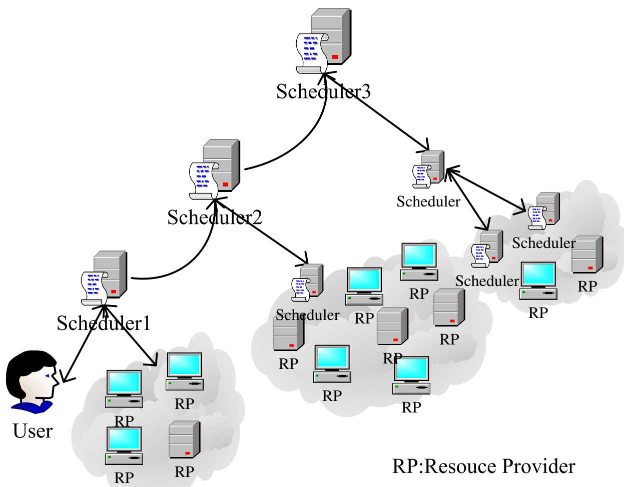

Figure 1: The Grid Market with Hierarchical Scheduler

3. Assign jobs to the resource providers in local domain, if some jobs are rejected, go to next step, otherwise terminate.

4. Transmit the rejected jobs to upper scheduler (the Scheduler2 in Fig. 1), the upper scheduler will forward these jobs to other sub-schedulers, search for resource providers from other domains, and assign jobs to them.

5. If all jobs have been assigned to resource providers successfully, then terminate, otherwise repeat step4 until arriving at the job's deadline.

3. Extended resource consumers' utility model

Let $J_i = (B_i, D_i, \overline{g}_i)$ be the description of job i $(1 \leqslant i \leqslant m)$, where B_i denotes the budget restrict, D_i denotes the deadline, and $\overline{g}_i$ denotes the satisfaction value which will be described in section 3.2.

3.1 Resource consumers' utility model

We use the formulation defined in [9] to describe the cost of allocating jobs to resources.

t_{ij}: the total processing time required for job i if assigned to the jth resource.

p_j: the price value of per unit computing power and second of the jth resource.

r_j: the computing power supplied by the jth resource.

$x_{ij} = 1$, if the job i is assigned to the jth resource successfully, otherwise 0.

L_i: the penalty cost when the job i isn't processed successfully.

The utility function is given below which is subject to deadline and budget restricts:

$$Minimize:$$

$$\sum_i \sum_j t_{ij} p_j r_j x_{ij} + \sum_i (1 - \sum_j x_{ij}) L_i \tag{2}$$

We would like an optimal solution, but this problem is proved strongly NP-hard [7][9][14], and does not have a polynomial time approximation schema. We are going to improve the HRED algorithm to get a near-optimal solution to this problem. We increase the scheduling efficiency with the resource selection mechanism, and improve reliability by quantifying risk to cost.

3.2. Resource selection

Resource selection only leaves those resource providers with lower price and better trust degree behind. Let g_i^j denote the satisfaction value placed on jth resource provider for the ith job, then all the satisfaction values can be expressed as $G_i = [g_i^1, g_i^2, ..., g_i^n]$ where n denotes the number of resource providers for job i. Calculate

$$g_i^j = \frac{\alpha}{p_j} + (1 - \alpha) c_i^j \tag{3}$$

where p_j denotes the price of the jth resource provider, c_i^j denotes the trust degree of the jth provider, and α denotes the weight assigned to the price dimension.

Table 1: Resource selection

Input:	The resource price value and trust degree of each resource provider, and user's job i.
Output:	Resource providers subset RP_i.
	for each rp_i^j $(1 \leqslant j \leqslant n)$ in current providers list
	get his price value p_i^j and trust degree c_i^j;
	calculate g_i^j (3);
	if $g_i^j > \overline{g}_i$ then
	join rp_i^j into RP_i;
	endif
	endfor
	return RP_i to HRED;

Resource selection (Table 1) returns a resource providers set which contains the best resource providers whose price is low and trust degree is high, meaning these providers offer better services. With the decrease of the number of providers, the scheduling rate is increased. Here, the size of the set should be appropriate corresponding to the jobs' number. The resource selection will not increase time cost because this process is executed when traversing all the resource providers, which will be executed on original HRED too.

3.3. Quantify risk to cost

When people submit their jobs to Grid, they suffer risk somewhat because of the resource providers' unreliability. Trust is one major concern for consumers, who may not want their applications mapped onto resources that are owned or managed by malicious entities. To secure their requirements, we enhance the HRED algorithm by quantifying trust degree to part of cost.

Example: Job A is assigned to resource provider B whose trust degree is 0.8. Given Job A's processing time is 10 seconds, and resource provider B price his resource at 1 dollar per unit of computing power and second, if B provides 10 units of computing power, then the converted cost of job A can be calculated like this:

$$\frac{10seconds \times 10units \times 1(dollar/second \cdot unit)}{0.8}$$
$$= 125\ dollars$$

Considering rp_i^j's trust degree c_i^j, we calculate user's utility as:

$$V_{ij} = (L_i - \frac{t_{ij} p_j r_j}{c_i^j}) \quad (4)$$

With the risk being converted to part of cost, the scheduled resources by HRED algorithm give not only a cost minimization solution, but also a credible solution. If considering the time factor, we can also extend this problem to time minimization problem easily by converting risk to part of time.

3.4. Resource scheduling with improved HRED heuristic

Based on the hierarchical scheduler (Fig. 1), the improved HRED will schedule resources from local domain firstly. When the number of jobs is large, and local providers aren't sufficient for these jobs, some jobs will be transmitted to other domains. Dividing resource providers into domains could give a better performance on the scheduling efficiency. The overall improved HRED algorithm is given below:

1. Local scheduler generates RP_i, in which resource providers are all from local domain, and have a $g_i^j > \overline{g}_i$, where $\overline{g}_i$ is the lower bound of job i's satisfaction.

2. Calculate the rank V_{ij} (4) for each combination of job i and resource j, and sort them in non-increasing order of V_{ij}, here $i = 1, 2, ..., m$, $j \in RP_i$.

3. Let β' be the job and γ' be the resource in the first combination of the updated sorted list, and η be the set of jobs that have already been assigned to γ'.

4. Combine all the jobs in set η with the job β' and sort all of them in non-decreasing order of their deadline and ready time. Now try to assign these jobs to resource γ' in the sorted order.

5. If the time constraint is satisfied for the combination of β' and γ', then assign β' to γ' and remove all the combinations of β' from the sorted list.

6. If the updated list is empty, then terminate. If a job is rejected by local scheduler, transmit this job to upper scheduler. Otherwise select the next combination and go to Step 4.

The improved HRED algorithm inherits all the advantages from original HRED which gives a good scheduling solution on multi-attribute resource allocation problem, and it also schedules faster, and providers higher success rates for the users. A simulated Grid market is given to evaluate the improved HRED algorithm in section 4.

4. Simulation

We evaluate the improved HRED scheduling algorithm in two aspects: scheduling efficiency and success rates. In all experiments, we set each resource provider a random trust degree $c_i^j \in [0, 1]$, and a random price value $p_j \in [1, 10]$.

First, we take a look at the scheduling efficiency. In this experiment, we take different number of nodes but the same trust situation. Fig. 2 illustrates the changes in scheduling efficiency. We use two groups of jobs, where 50 jobs in the first group and 100 jobs in the second group. Through the Fig. 2, we can see that when there are large numbers of resource providers in the market Grid, compared with original HRED, the improved HRED spend much less time in scheduling, this is because the resource selection and hierarchical scheduler mechanism reduce the number of resource providers and computing. As illustrated from Fig. 2, we also note that it takes a little more time to schedule 100 jobs than to schedule 50 jobs.

In addition to providing a faster scheduling speed, the improved HRED also leads to a higher success rate, whose value is above 0.8 when there are 200 or even more resource providers in the grid market (Fig. 3). In contrast, if we do not consider the trust situation in HRED, because we just give a random trust degree, the success rates just fluctuate around at the 0.5.

Thereafter, we take a look at the success rates when different $\overline{g}_i$ values are applied. In Fig. 4, we submit 50 jobs to

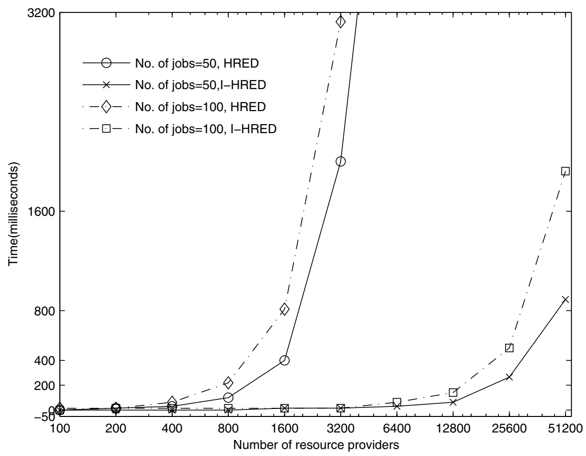

Figure 2: The scheduling time along with different number of resource providers

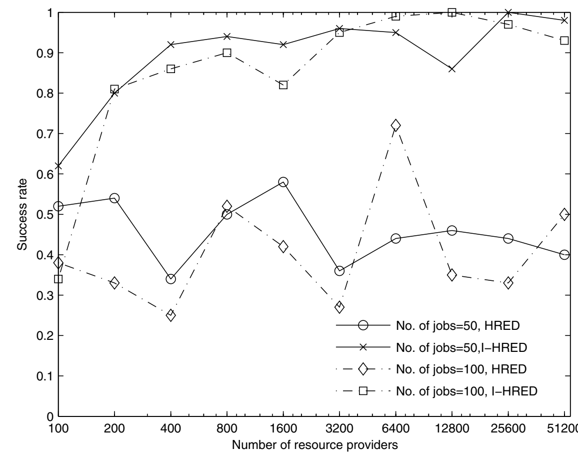

Figure 3: Success rates along with different number of resource providers

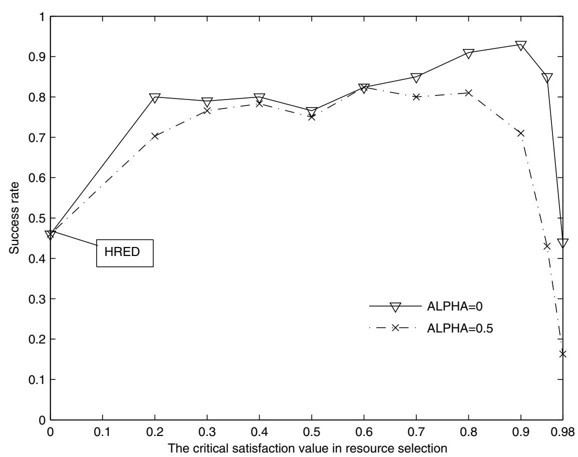

Figure 4: Success rates with different $\overline{g}_i$ value

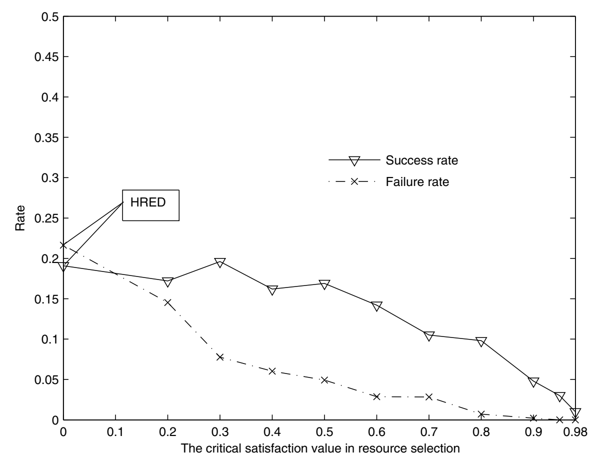

Figure 5: Success rates and failure rates when large numbers of jobs are submitted

100 resource providers, and we set the job i's budget $B_i \in [8000, 10000]$, the submit time $Submit\_time_i \in [0, 20]$, deadline $D_i \in [0, 120]$, resources $r_j \in [800, 1000]$, and L_i is large enough depending on each job's cost.

Let $\alpha = 0$, which means the resource selection only cares about providers' trust degree, along with the increase of $\overline{g}_i$, the success rate rises until $\overline{g}_i > 0.9$, then because the number of available resource providers decreases for the too high $\overline{g}_i$ value, the success rate declines too. If trust degree isn't considered (It means $\overline{g}_i = 0$), the success rate value is around 0.5 because of the random trust degree. Then let $\alpha = 0.5$, we find that the success rate declines a little earlier, for the number of resource providers that have both low price and a high trust degree is less than when $\alpha = 0$.

In Fig. 5, we try to submit 500 jobs to 10 resource providers while other parameters are the same with Fig. 4. We can see that the success rate declines along with the increase of $\overline{g}_i$ value because of the available resource providers' decrease. However, the failure rate which means failure jobs' ratio caused by resource providers' malicious behavior also decreases, meaning that most of the failures are caused by lack of resource providers rather than malicious behavior.

5. Conclusion and future work

Compared with original HRED algorithm, our proposal does not only provide resource consumers with a fast and credible mechanism to schedule resources, but also inherits the advantages from original HRED that gives a near-optimal solution on cost minimization scheduling problem, and it can also be extended to time minimization easily. Our experiments shows that the improved HRED algorithm gives a good success rate and scheduling efficiency especially when the number of resource providers is large.

Applying economic principles to Grid is deemed promising to improve the system overall value [6]. However, the Grid market can only be approximated as a perfect competitive market at best, and it is hard to get into the equilibrium state in reality. So the economic mechanism, including pricing policy and currency mechanism, should be designed properly to meet the both consumers and producers' utility, while improving the overall value. The next step, we will pay much more attention on the pricing policy and currency mechanism.

6. Acknowledgment

The paper is supported by National Nature foundation of China under grant No.s: 60673046, 90715037, 60805024, by Specialized Research Fund for the Doctoral Program of Higher Education (SRFDP) under grant No.: 200801410028, by National 973 Plan of China under grant No.: 2007CB714205 and by Natural Science Foundation Project of Chongqing, CSTC under grant No.: 2007BA2024.

We are grateful to Hui Zhao and Cheng Guo for the constructive comments and suggestions that significantly improved the paper.

References

[1] A. AuYoung, B. Chun, A. Snoeren, and A. Vahdat. Resource allocation in federated distributed computing infrastructures. *Proceedings of the 11th Workshop on Hot Topics on Operating Systems and Architectural Support for the On-demand IT InfraStructure*, October 2004.

[2] R. Buyya. Economic-based distributed resource management and scheduling for grid computing. *Ph.D. Thesis*, Monash University, Melbourne, Australia, April 2002.

[3] R. Buyya, D. Abramson, and J. Giddy. Nimrod/g: An architecture for a resource management and scheduling system in a global computational grid. *Proceedings of the International Conference on High Performance Computing in Asia-Pacific Region*, 1:283–289, May 2000.

[4] R. Buyya, M. Murshed, D. Abramson, and S. Venugopal. Scheduling parameter sweep applications on global grids:a deadline and budget constrained cost-time optimization algorithm. *Software: Practice and Experience (SPE) Journal*, 35:491–512, April 2005.

[5] R. S. Chang, J. S. Chang, and P. S. Lin. An ant algorithm for balanced job scheduling in grids. *Future Generation Computer Systems*, 25(1):20–27, January 2009.

[6] J. Cheng and M. Wellman. The walras algorithm: A convergent distributed implementation of general equilibrium outcomes. *Computational Economics*, 12(1):1–24, August 1998.

[7] T. Eymann, M. Reinicke, and W. S. et al. Catallaxy-based grid markets. *Multiagent and Grid Systems*, 1(4):297–307, December 2005.

[8] D. E. Irwin, L. E. Grit, and J. S. Chase. Balancing risk and reward in a market-based task service. *Proceedings of 13th IEEE International Symposium on High Performance Distributed Computing (HPDC)*, pages 160–169, June 2004.

[9] S. Kumar, K. Dutta, and V. Mookerjee. Maximizing business value by optimal assignment of jobs to resources in grid computing. *European Journal of Operational Research*, 194(3):856–872, May 2009.

[10] K. Lai, L. Rasmusson, E. Adar, L. Zhang, and B. Huberman. Tycoon: an implementation of a distributed, market-based resource allocation system. *Multiagent and Grid Systems*, 1(3):169–182, August 2005.

[11] N. G. Mankiw. *Principles of Economics,3rd ed*. South-Western College Publisher, 2003.

[12] S. Song, K. Hwang, and Y. Kwok. Risk-resilient heuristics and genetic algorithms for security assured grid job scheduling. *IEEE Transactions on Computers*, 55(6):703–719, June 2006.

[13] K. Subramoniam, M. Maheswaran, and M. Toulouse. Towards a micro-economic model for resource allocation grid computing systems. *The 2002 IEEE canadian conference on Electrial & Computer Engineering*, 2(1):782–785, August 2002.

[14] J. Yu and R. Buyya. Scheduling scientific workflow applications with deadline and budget constraints using genetic algorithms. *Scientific Programming*, 14(3):217–230, December 2006.