2008 International Conference on Computer Science and Software Engineering

Market-driven Based Resource Scheduling Algorithm in Computational Grid

Bing Tang, Zude Zhou, Quan Liu, Fangmin Li
School of Information Engineering
Wuhan University of Technology
Wuhan, 430070, China
Email: tangbing@whut.edu.cn

***Abstract*—Grid resource management and scheduling are confronted with a great challenge. This paper presented a dynamic resource price-adjusting (RPA) strategy in computational Grid. Three new concepts and two new evaluation standards are defined. On the basis of these theories, a Grid resource scheduling algorithm based on marker-driven using dynamic RPA strategy is proposed. With the help of GridSim toolkit, the simulation of resource scheduling algorithm proposed in the paper is realized, and the new algorithm is compared with GridSim DBC (deadline and budget constrained) scheduling algorithms. The experimental results show that the new algorithm improves the performance of Grid in aspect of increasing the task accomplishment ratio and solving the problem of load imbalance.**

Keywords–Grid computing; resource scheduling; market-driven.

I. Introduction

The Grid is a name that was first coined in the mid-1990s to denote a proposed distributed computing infrastructure for advanced science and engineering projects. Grid should enable coordinated resource sharing and problem solving in dynamic and multiple virtual organizations [1] [2] [3]. Resource sharing is the essence of Grid. There are a wide range of heterogeneous and geographically distributed resources in Grid, and they are with different capabilities and configurations, and Each Grid user or consumer has their own QoS requirements. In a word, the resource management and scheduling in the Grid environment are confronted with a great challenge.

Resource scheduling algorithm is a popular issue in Grid research field, and there are many research papers about resource scheduling algorithm reported in recent years. Current resource scheduling algorithms are mainly based on Min-Min [4], Max-Min, heuristic algorithm, genetic algorithm (GA), ant algorithm (AA), bid mechanism, auction mechanism [5], computational economy model [6] [7] [8] [9], and so on.

Among these algorithms, few algorithms have considered about the factor of cost and time. As we all know, cost and time are both important QoS attributes [10]. Hence, most of current algorithms cannot provide comprehensive QoS assurance, which results in Grid jobs are scheduled to execute on a resource that maybe not satisfy the QoS requirements. Another shortcoming of current algorithms is that most of them have not solved the problem of load imbalance. Some resources may be load overweight and others may have nothing to do with an empty job list. In order to improve the load imbalance problem of traditional computational economy model [6] [7] [8] [9], this paper proposed a market-driven based resource scheduling algorithm.

II. Market-driven Based Grid Resource Scheduling Algorithm

A. Market Theory

Due to the characters of Grid resource and Grid user, economics theory is widely applied in Grid research fields, especially Grid resource management filed. Market mechanism can be imported to solve the resource distribution and scheduling problem in Grid, where that Grid is regarded as a virtual market, and Grid resource is regarded as commodity, and commodity has its price, and competition exists in the market [5] [11] [12]. Due to the similarities between Grid and economical market, economic-based model is fit for solving Grid resource management problem.

In resource management system based on economy model, resource provider is defined as producer, and resource user is defined as consumer. Producers and consumers become individuals in economic environment, and each individual has their own target and strategy. Market-driven resource management and scheduling system should maximize profit of each individual. The strategy of consumer is to use lowest expense to solve his problem, while the strategy of producer is to attract more consumers and makes his profit maximum.

Resource owners or provider determine the price of their resources, and charge customers for consuming the resource. Pricing strategy is that price depends on some factors, and price can be fixed or fluctuates as the changing of supply and demand.

B. Resource Price-adjusting Mechanism

DBC scheduling algorithms with Cost-optimization and Time-optimization provided by GridSim is usually compared with other new scheduling algorithms based on price, cost and time [7]. Simulation results reveal and prove some weaknesses of DBC algorithms, which have weakness of load imbalance.

The rule of supply and demand is the theoretical foundation of economics, it is also called market rule. In the paper, a resource price-adjusting (RPA) mechanism is proposed, by applying this rule in Grid resource scheduling, which enables the resource adjusting its price according to its load condition. The price dynamically adjusting influences the behavior of

978-0-7695-3336-0/08 $25.00 © 2008 IEEE
DOI 10.1109/CSSE.2008.996

user broker choosing resource, hence resource load balance in the grid system is achieved, and task accomplishment ratio is increased.

Here are some related definitions and conceptions in RPA mechanism.

The price of a Grid resource is dynamically adjusted according to the following equation:

$$p = \begin{cases} \lambda_1(l - l_0) + minp & (0 \le l \le b) \\ \lambda_1(b - l_0) + \lambda_2 e^{l-b} + minp & (l > b) \end{cases} \quad (1)$$

where l_0 denotes the local load of Grid resource; l denotes the current total load of Grid resource; b signifies a threshold which controls the price adjusting in different ways; λ_1 and λ_2 control the range of price fluctuating.

We define the maximal price, the minimal price and the price range of a Grid resource, as you see in (2).

$$\begin{array}{l} minp = p_0(1 - \alpha) \\ maxp = p_0(1 + \alpha) \\ minp \le p \le maxp \end{array} \quad (2)$$

where p denotes the price of Grid resource; p_0 denotes the central price of Grid resource, and it is fixed by the owner of resource; denotes the fluctuant factor of price, which describes the price range, and it is also given by the owner.

C. Three Definitions and Two Evaluation Standards

Definition 1: *The Life-time of Price*

The price of resource is dynamically adjusted, and the time of the price value lasted called as the life-time of the price. If the broker submitted a new job to the resource and the resource need adjust to a new price, but there is a job running on the resource and not finished, and then the cost of the unfinished job is still calculated according to the old price. After the unfinished job accomplished, the new price becomes effective. We define that the initial price of Grid resource in simulation experiment is its central price p_0.

Definition 2: *The Load Factor of Grid Resource*

The load factor of Grid resource is evaluated by the length of job queue submitted to the resource. Longer the job queue is, busier the resource is. As you see in (3), the load factor is marked as lf.

$$lf = \frac{l_{jq}}{n_{pe} m_{pe}} \quad (3)$$

where l_{jq} is denoted as the length of job queue; n_{pe} is denoted as the number of PEs of the resource; m_{pe} is denoted as the process speed of one PE (evaluated as MIPS or SPEC).

Definition 3: *The Load Balance Factor of Grid System*

The performance evaluation standard of load balance for Grid system is defined as (4). If the value of load balance factor for Grid system is small, it means that the load of Grid system is balanced.

$$Q = \sqrt{\sum_{i=1}^{n} \left(lf_i - \overline{lf}\right)^2} \quad (4)$$

where we define $\overline{lf}$ as the average load factor of all resources in Grid system, which is showed as follows:

$$\overline{lf} = \frac{1}{n} \sum_{i=1}^{n} lf_i \quad (5)$$

The new scheduling algorithm based on Grid resource price-adjusting mechanism allows Grid user set resource selection strategy according to their requirements. This paper studied two strategies, one is Cost-optimization aiming at reducing cost, and the other is Time-optimization aiming at reducing task execution time.

III. Algorithm Description

A. RPA-Cost Optimization Algorithm

1) RESOURCE DISCOVERY.
2) UPDATE PRICE of resources according to price-adjusting equation.
3) RESOURCE TRADING: Identify new CPU cost per time unit and capacity to be delivered per cost-unit of each resource.
4) SORT resources by increasing order of CPU cost per second.
5) SCHEDULING: Repeat while there exists unprocessed jobs in application job list and the time and process expenses are within deadline and budget limits. It is triggered for each scheduling event or whenever a job completes. Redo UPDATE PRICE and RESOURCE TRADING before each trigger. SCHEDULE Policy is defined as:
    a) For each resource perform load profiling to establish the job consumption rate or the available resource share through measure and extrapolation.
    b) For each resource based on its job consumption rate or available resource share, predict and establish the number of jobs a resource can process by the deadline.
    c) For each resource in order:
        i) If the number of jobs currently assigned to a resource is less than the predicted number of jobs that a resource can consume, assign more jobs from unassigned job queue or from the most expensive machines based on job state and feasibility. Assign job to a resource only when there is enough budget available.
        ii) Alternatively, if a resource has more jobs than it can complete by the deadline, move those extra jobs to unassigned job queue.
6) DISPATCHER Policy: Repeat the following steps for each resource if it has jobs to be dispatched: Identify the number of jobs that can be submitted without overloading the resource. The policy is to dispatch jobs as long as the number of user jobs deployed (active or in queue) is less than the number of PEs in the resource.

B. RPA-Time Optimization Algorithm

1) RESOURCE DISCOVERY.
2) UPDATE PRICE.
3) RESOURCE TRADING.
4) SCHEDULING: SCHEDULE Policy is defined as:
    a) For each resource, predict and establish the job consumption rate or the available resource share through the measure and extrapolation strategy taking into account the time taken to process previous jobs.
    b) If any of the resource has jobs assigned to it in the previous scheduling event, but not dispatched to the resource for execution and there is variation in resource availability, then move appropriate number of jobs to the Unassigned-Jobs-List. This helps

in updating the whole scheduling based on the latest resource availability information.

c) Repeat the following steps for each job in the Unassigned-Jobs-List:

   i) Select a job from the Unassigned-Jobs-List.
   ii) Create a resource group containing affordable resources whose processing price is less than or equal to the remaining budget per job.
   iii) For each resource in the resource group, calculate/predict the job completion time taking into account previously assigned jobs and the job completion rate and resource share availability.
   iv) SORT resource in the resource group by the increasing order of job completion time.
   v) Assign the job to the first resource in the resource group and remove it from the Unassigned-Jobs-List if the predicted job completion time is less than the deadline.

5) DISPATCHER Policy.

IV. Simulation Experiments

A. Experiment Setup

(1) Resource modeling

We modeled and simulated 9 time-shared resources and 1 space-shared resource with different characteristics. From the Table I, the 10 resources can be divided into 3 types according to their processing speed, as follows:

a) Advanced resources, including: R1, R4, R5, R9 and R10;
b) Medium resources, including: R6 and R7;
c) Low-class resources, including: R2, R3 and R8;

Considering price and performance synchronously, we may figure out that R3, R4, R5 and R10 are better resources, due to they have a high computing capability per cost-unit.

(2) Application modeling

We have modeled a task farming application that consists of 200 jobs, so that each user submits 200 jobs to the broker. In GridSim, each job is packaged as Gridlet, and the Gridlet's job length (processing requirements) is defined as 50000 MI (million instructions) with a random variation of 0 to 10% on the positive side. The length of input file and output file is ignored.

In the simulation, we performed scheduling experiments with different values for a single user. The deadline is varied in simulation time from 100 to 5100 in steps of 500. The budget is varied from G$20000 to 70000 in steps of G$5000. We performed scheduling simulation for the Cost-Optimization algorithm and Time-Optimization algorithm which are provided by GridSim, and we also evaluate RPA-Cost Optimization algorithm and RPA-Time Optimization algorithm which are proposed in this paper.

TABLE I
Attributes and Characteristics of Grid Resources.

Resource ID	No. of PEs	MIPS of each PE	Manager Type	Price (G$ /time unit)	MIPS Rating (/G$)
R1	32	2200	Time-shared	12	183.33
R2	8	377	Time-shared	2	188.50
R3	4	377	Time-shared	1	377.00
R4	16	2000	Time-shared	10	200.00
R5	8	1600	Time-shared	8	200.00
R6	8	766	Time-shared	5	153.20
R7	8	800	Time-shared	5	160.00
R8	16	410	Space-shared	3	136.67
R9	64	2200	Time-shared	15	146.67
R10	16	1800	Time-shared	9	200.00

TABLE II
Results of Cost-Optimization.

Deadline	Budget	Number of Gridlets Completed	Load Balance Factor	Resource Used
2600	45000	200	4.5032	R3, R4
100	55000	188	0.1448	R1, R4, R5, R9, R10
3100	35000	174	5.5136	R3, R4
5100	70000	200	9.0421	R3, R4

TABLE III
Results of Time-Optimization.

Deadline	Budget	Number of Gridlets Completed	Load Balance Factor	Resources Used
2600	45000	200	4.0397	R1, R3, R4, R5, R10
100	55000	80	0.1633	R4, R5, R10
3100	35000	88	5.5361	R3
5100	70000	200	1.5947	R3, R4, R8, R9, R10

B. GridSim DBC Scheduling Review

Table II and Table III shows the selection of resources for processing Gridlets for different budget values with a fixed deadline of 100, 2600 3100, and 5100 (low, medium and high values, respectively). We make a conclusion that the whole system is not load balanced, and most of the resources at most of the time are free.

We just choose four typical conditions, and list their results in terms of the number of Gridlets completed, load balance factor, and resource used. The four conditions are as follows:

a) deadline=2600 and budget=45000;
b) deadline=100 and budget=55000;
c) deadline=3100 and budget=35000;
d) deadline=5100 and budget=70000.

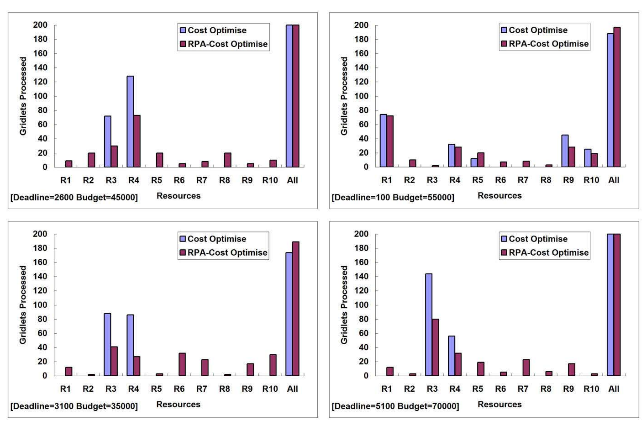

Fig. 1. Number of Gridlets completed in each resource with Cost-Optimization and RPA-Cost Optimization.

C. RPA strategy Simulation Results

Resource scheduling algorithm allows price dynamically adjusting, and each resource must set a central price and a

TABLE IV
CENTRAL PRICE AND FLUCTUATE FACTOR OF EACH RESOURCE.

Resource ID	Central Price (G$)	Price Fluctuate Factor	Maximal Price (G$)	Minimum Price (G$)
R1	12	0.30	15.60	8.40
R2	2	0.75	3.50	0.50
R3	1	0.80	1.80	0.20
R4	10	0.40	14.00	6.00
R5	8	0.40	11.20	4.80
R6	5	0.60	8.00	2.00
R7	5	0.60	8.00	2.00
R8	3	0.75	5.25	0.75
R9	15	0.20	18.00	12.00
R10	9	0.40	12.60	5.40

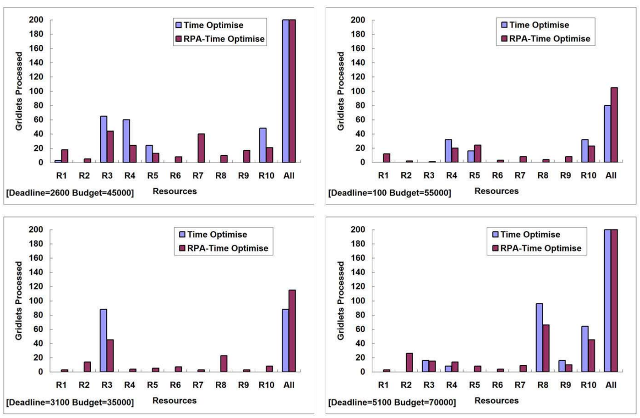

Fig. 2. Number of Gridlets completed in each resource with Time-Optimization and RPA-Time Optimization.

fluctuant factor, and the range of price is determined by central price and fluctuant factor according to (2). In the simulation, attributes and characteristics of Grid resources are similar to Table I, The only difference lies in the existence of central price, fluctuant factor, maximal price and minimal price. Table IV shows the new attributes of Grid resources.

Cost-Optimization is compared with RPA-Cost Optimization algorithm in terms of the number of Gridlets completed (see Figure 1) and the load balance factor of Grid system (see Table V) in the four conditions (the same as the preceding section). From Figure 1 and Table V, we are aware that in the four conditions, the value of load balance factor with RPA-Cost Optimization is smaller than with Cost-Optimization, and RPA-Cost Optimization algorithm improves the problem of load unbalanced, and the number of completed Gridlets increases.

TABLE V
RESULTS OF RPA-COST OPTIMIZATION.

Deadline	Budget	Number of Gridlets Completed	Load Balance Factor
2600	45000	200	1.8152
100	55000	197	0.1181
3100	35000	189	2.5034
5100	70000	200	4.9452

TABLE VI
RESULTS OF RPA-TIME OPTIMIZATION.

Deadline	Budget	Number of Gridlets Completed	Load Balance Factor
2600	45000	200	2.6764
100	55000	105	0.1506
3100	35000	115	2.7612
5100	70000	200	1.1969

RPA-Time Optimization algorithm is compared with Time-Optimization in the four conditions, and the comparison results are shown in Figure 2 and Table VI, which indicate RPA-Time Optimization algorithm increases the task accomplishment ratio and solves the problem of load imbalance.

V. CONCLUSION

The economy-based market model for large-scale applications in computational Grid is discussed. RPA-Cost Optimization and RPA-Time Optimization algorithm are proposed in this paper. These two improved scheduling algorithm are based on a dynamic price-adjusting strategy, which introduces the concept of central price, the fluctuant factor of price, and the lifetime of price; the performance evaluation standard of load balance for Grid system is defined using a load balance factor. Extensive simulation experiments with different value of deadline and budget for a single user demonstrate the advantages of RPA-based scheduling algorithm in terms of increasing the task accomplishment ratio and solving the problem of load imbalance.

ACKNOWLEDGMENT

This paper is supported by Major International Cooperation and Exchange Program of National Natural Science Foundation of China under Grant No. 50620130441.

REFERENCES

[1] I. Foster and C. Kesselman, “Globus: a metacomputing infrastructure toolkit,” *International Journal of High Performance Computing Applications*, vol. 2, pp. 115–128, 1997.

[2] I. Foster, *The Grid 2: Blueprint for a New Computing Infrastructure*. USA: Morgan Kaufmann, 2003.

[3] I. Foster, C. Kesselman, and S. Tuecke, “The anatomy of the grid: enabling scalable virtual organizations,” *International Journal of High Performance Computing Applications*, vol. 3, pp. 200–222, 2001.

[4] X. S. He, X. H. Sun, and G. V. Laszewski, “Qos guided min-min heuristic for grid task scheduling,” *Journal of Computer Science & Technology*, vol. 3, pp. 442–451, 2003.

[5] B. Pourebrahimi, “Market-based resource allocation in grids,” in *Proceedings of the 2nd IEEE International Conference on e-Science and Grid Computing*, 2006.

[6] R. Buyya, S. Chapin, and D. DiNucci, “Architectural models for resource management in the grid,” in *Proceedings of the 1st IEEE/ACM International Workshop on Grid Computing*, 2000, pp. 18–35.

[7] R. Buyya and M. Murshed, “Gridsim: a toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing,” *Concurrency and Computation: Practice and Experience*, vol. 14, pp. 1175–1220, 2002.

[8] D. A. R. Buyya and S. Venugopal, “The grid economy,” *Proceedings of the IEEE*, vol. 3, pp. 698–714, 2005.

[9] O. O. Sonmez and A. Gursoy, “A novel economic-based scheduling heuristic for computational grids,” *International Journal of High Performance Computing Applications*, vol. 1, pp. 21–29, 2007.

[10] R. Al-Ali, “Qos support for high-performance scientific grid applications,” in *Proceedings of 2004 IEEE International Symposium on Cluster Computing and the Grid*, 2004, pp. 134–143.

[11] M. F. Ma, “An fair model for computing economy of grid resource management,” *Microelectronics and Computer*, vol. 1, pp. 28–35, 2006.

[12] X. W. Wang, “A microeconomics-based resource assignment model for grid computing,” *Journal of Northeastern University*, vol. 7, pp. 731–734, 2006.