A Quality-Driven Algorithm for Resource Scheduling Based on Market Model on Grid

Lei Tang[1], Zhiyi Yang[1], Zhiwen Yu[2],Yunlan Wang[1]
Center of High Performance Computing, Northwestern Polytechnical University
Xi'an 710072, China
Information Technology Center, Nagoya University
Furo-cho, Chikusa-ku, Nagoya 464-8601, Japan
tanglei24@mail.nwpu.edu.cn

Abstract

Several challenges about computational grid exist in integrating, coordinating and managing of resources and scheduling of applications, due to distributed resources at various levels. In this paper, we employ computational economy as a way to manage and allocate these distributed resources. It can help to rapidly and dynamically select resources meeting users' functional requirements and satisfying their Quality of Service (QoS), from a large number of resources. An economic scheduling system based on market model is presented. On the basis of studying QoS criteria of resource services, we formulate selection as Multiple Attribute Decision Making (MADM) problem. Then, we introduce MQoS model to evaluate resource services for selection, which consists of the MQoS vector, a decision matrix and a quality-driven scheduling algorithm. The algorithm is classified as lexicographic and Euclidean Distance algorithm with respect to user's preference and object impact. It optimizes system performance through adjusting resource selection automatically according to previous job execution and the current system state information such as load. The results of scheduling experiments showed the superiority of quality-driven algorithm in scheduling on global grids.

1. Introduction

Grid resource management among which scheduling is an important part is challenging due to several issues desired. Some of these issues are: (a) supporting distributed and heterogeneous resources; (b) allowing resources with different administrative policies while preserving owner autonomy; (c) allowing resources with the availability and capability changing dynamically; (d) providing the uniformed description of resources; and (e) supporting user' quality of service requirements. Based on above issues, a grid scheduler differs from a scheduler for conventional computing systems in several aspects. One of the primary differences is that the local resources are in general, not controlled by the global grid scheduler, but by the local scheduler. Another is that the grid scheduler cannot assure that it is the optimal scheduling from a global view of the grid. We are interested in the problem of scheduling on grid, which is significant to consider the individual preferences of users and the dynamicity of resources.

Some metacomputing systems such as Globus [1] and Legion [2] adopt queue system to manage high-end resource centrally, providing an infrastructure used in distributed high-end application. It is unavailable to construct a computational grid and provide dynamic and flexible resource scheduling, as the centralized strategy such as Min-min, Max-min [3, 4], could not obtain the complete state information and achieve common management strategies in global grid system. More specifically, the current grid schedulers with various scheduling algorithms have difficulty in guaranteeing the quality of schedules they produce. The most challenging issue that they encounter is the dynamicity of resources. Although a resource may be participating in a grid, its main purpose is for use by local users of the organization that it belongs to. Therefore, the load on the resource imposes a great strain on grid scheduling.

With the proliferation of grid, at least one new things need to be considered in a scheduling model. That is the quality of service. The scheduler in the grid environment needs to consider QoS to get a better match between applications and resources. In this study, we embed the QoS information into the scheduling algorithm to improve the efficiency and the utilization of a grid system. From the viewpoint of QoS, the previous scheduling algorithms only consider the local system performance where users are passive to accept services and they cannot bring forward the additional requirement, so they have neglected the variety of user's service demand.

Whereas, investigating the use of computational market for automating trades between different self-interested parties, many researchers have realized that market model has emerged as a new way to solve the resource management challenges. First of all, market mechanism reflects the supply and demand of resources through

2007 International Conference on Parallel Processing Workshops (ICPPW 2007)
0-7695-2934-8/07 $25.00 © 2007 IEEE

dynamic price fluctuation and optimizes the resource thorough the equilibrium of supply and demand; secondly, the dynamic coordinated resource management mechanism also suits the grid resources characteristic. Currently available scheduling algorithms based market model have be proposed. Buyya and Giddy [5] present several optimization scheduling algorithms (DBC) under the limitation of time and cost in the Nimrod-G model. The application level resource broker for the grid, has incorporated three adaptive deadline and budget constrained scheduling based on QoS criteria, e.g. execution time and execution cost.

DBC algorithm, which is based on the application level QoS scheduling, has satisfied the demand of user from different aspects. Whilst in actual Grid environment, user's demands and applications are different. The scheduling algorithm only balance time and cost comprehensively with neglecting the variety of user’s QoS requirements.

Therefore, on the basis of grid economical model proposed by Buyya and Giddy [5], we propose a scheduling system, in which resource selection is described by MQoS model. It also supports quality-driven scheduling algorithm that improves DBC algorithm in terms of meeting multiple QoS demands. The system introduces credit criterion to evaluate trustworthiness of each QoS criterion of a resource in order to reflect the dynamicity of resources.

The rest of the paper is organized as follows. Section 2 gives a scheduling system based on market model. Section 3 presents a MQoS model for evaluating resource services and its formal definition. Section 4 formulates quality-driven scheduling algorithm according to users’ preference and objectivity impact. Sample results of the algorithm are discussed in Section 5. The final section summarizes the paper along with suggestions for future works.

2. Scheduling system based on market model

In this paper, market model is considered as a basic model, where resource owners specify their service price and charge users according to the amount of resource they consume. The pricing policy can be derived from various parameters based on Grid Service Providers’ (GSPs) desired profit margin and users’ demand, and can be variable depending on the resource supply and demand [6]. We apply a new scheduling system as shown in Fig. 1 that supports preference-oriented and objective-based scheduling of applications on the computational grid. Depending on users’ QoS requirements, our scheduling system dynamically leases grid services at runtime depending on their cost, quality, and availability. The scheduler can provide the ability to specify resources, obtain quick turnaround for user, and receive reliable allocation of resources with all sorts of information gathered by a resource discoverer.

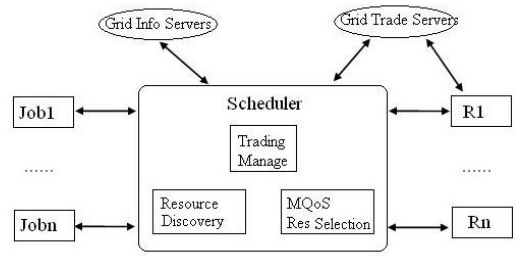

Figure 1: Scheduling system based on market model

The scheduler is responsible for resource discovery, resource trading, resource selection, and job assignment. The resource discovery algorithm interacts with a grid-information service (GIS) directory, identifies the list of authorized machines, trades for resource access cost, and keeps track of resource status. The resource selection is responsible for selecting those resources that meet the deadline and requirements, and minimize the cost of computation.

To support rapid and effective selection, resources that meet users’ functional requirements must be able to be located from a large number of resource owner based on their Quality of Service (QoS), such as reliability, price, availability, etc. The scheduling system needs these resource properties to arrive at a quality-driven global resource selection based on QoS criteria and users’ preference. On the basis of analysis, we can regard the resource selection in grid as a hybrid MADM (Multiple Attribute Decision Making) problem with both quantitative and qualitative attributes and uncertainty weights of attributes. Thus, we present a selection model of QoS criteria called multi-dimension Quality of Service (MQoS) for resource, which allows QoS criteria to be specified as part of a resource service interface – to enable resource selection based on QoS.

The MQoS model is a part of scheduling system. It supports the discovery and selection of suitable resources, trading with resource owner, according to QoS criteria. Job execution and management rely on the existing systems such as Globus.

3. MQoS Selection Model

In service-oriented infrastructure of computing grid, GIS stores different QoS information used to describe services provided by different resources. To differentiate each resource, it is necessary to define the expression about resources. Consequently, a well-defined model of QoS criteria named MQoS for selecting resources is needed.

The model consists of three parts: MQoS vector, decision matrix and quality-driven scheduling algorithm.

3.1 MQoS vector

QoS criteria focus on the application level in this paper, such as delay, the reliability of data transmission and

2007 International Conference on Parallel Processing Workshops (ICPPW 2007)
0-7695-2934-8/07 $25.00 © 2007 IEEE

credit evaluation. In order to express resource uniformly, we employ a vector, called *MQoS* to describe the various physical (e.g. CPU) and social (e.g. evaluation) attributes of resource in varied application environments. To differentiate clearly, we divide the criteria of MQoS model into two types: one is benefit criteria and the other is cost criteria. This paper introduces a universal quantization MQoS vector to denote grid resources.

Definition 3.1 (Resource Set): R= $\{R_1, R_2, ..., R_n\}$, where R_i is a resource of the system. The resource information is derived from GIS.

Definition 3.2 (MQoS Vector): $MQoS = (QoS_1, QoS_2, ..., QoS_m)$, which can be customized by users in accordance with their applications, where QoS_i is the i^{th} QoS criteria of a resource R. New criteria can be added without fundamentally altering the resource selection techniques built on top of it.

Furthermore, we introduce credit criterion to evaluate trustworthiness of a resource, which mainly depends on user's experiences.

Definition 3.3 (credit criteria): A credit criteria of a resource R is the average marks on jobs execution on R, denoting as $QoS_{credit}(R)$. It is computed through the expression, $QoS^{j}_{credit}(R_i) = job.cputime / job.wallclocktime$, where $QoS^{j}_{credit}(R_i)$ is the i^{th} resource R have been marked by the j^{th} time, *cputime* is job running time on R_i, *wallclocktime* is job execution time on R_i, including the hanging up time due to some unexpected reasons and cputime. To enable the relevance of credit criterion and effective evaluation for resource performance, we synthesize the N times results as following, used as the credit criterion, $QoS^{N}_{credit}(R_i) = a \times \tan(\sum_{j=1}^{N} QoS^{j}_{credit}(R_i) / N)$, where N is the number of times the i^{th} resource R have been marked.

3.2 Decision Matrix

Definition 3.4 (Decision Matrix): M=$[q_{ij}]$, represents the decision matrix of QoS criteria as shown in Table 1, where q_{ij} denotes a numerical value about the j-th QoS criteria of the i-th resource.

Table 1: decision matrix M

alternatives (R)	quantitative attribs			qualitative attribs	
	QoS_1	QoS_2	…	…	QoS_m
R_1	q_{11}	q_{12}	…	…	q_{1m}
…	…	…	…	…	
R_n	q_{n1}	…	…	…	q_{nm}

In computing grid, resource selection can be formulated as a MADM problem with both quantitative and qualitative attributes. It is to rank the resources or select the best compromise R_i from them with both quantitative and qualitative attributes being simultaneously satisfied to the extent possible.

To deal with such a hybrid selection problem, a qualitative QoS attribute needs to be measured at first. To do so, a few evaluation grades may be defined, to which the state of the attribute may be evaluated. Based upon the evidential reasoning (ER) method as described in [7], the state of a qualitative attribute at each resource in Table 1 can be transformed into the numerical with the following steps.

1) Calculate the basic probability assignments for evaluation of qualitative attribute $QoS_j(R_i)$ from the given confidence degrees which represents uncertainty in the evaluations by using the formula,

$$m_j^n(R_j) = w_j \beta_n(R_j) \qquad (1)$$

$m_j^n(R_j)$ expresses a basic probability assignment to the QoS_j attribute at R_i. It is confirmed to the evaluation grades H_n from a set H= $\{H_1,\ldots,H_N\}$. $\beta_n(R_j)$ denotes a confidence degree associated with the state of QoS_j at resource R_i, being evaluated to H_n.

2) Quantify qualitative attribute value,

$$p_{rk} = \sum_{n=1}^{N} m_j^n(R_j) p(H_n) + (1 - \sum_{n=1}^{N} m_j^n(R_j) p(H_n)) \times p(H) \qquad (2)$$

where $p(H_n)$ represents the scale of H_n, which is set to be a real number in the closed interval [-1,1], and $p(H) = \sum_{n=1}^{N} p(H_n)/N$. The evaluation grade set is quantified by p{H} = $[p(H_1) \ldots p(H_n) \ldots p(H_N)]^T$, where $p(H_1)=-1$, $p(H_N)=1$, and $p(H_{n+1}) > p(H_n)$.

To avoid the different dimensions of quantitative attributes at each resource, matrix M needs to be normalized into a matrix N= $[n_{ij}]$ as follows, that is to put all values q_{ij} in a range from 0 to 1.

$$n_{ij} = \begin{cases} \dfrac{q_{ij} - \min\limits_i(q_{ij})}{\max\limits_i(q_{ij}) - \min\limits_i(q_{ij})} \text{.....for benefit criteria } QoS_j \\ \dfrac{\max\limits_i(q_{ij}) - q_{ij}}{\max\limits_i(q_{ij}) - \min\limits_i(q_{ij})} \text{......for cost criteria } QoS_j \end{cases} \qquad (3)$$

With the above steps being done, an extended decision matrix is then transformed into an ordinary normalized decision matrix, which can be dealt with by quality-driven scheduling algorithm.

4 Quality-driven scheduling algorithm

MQoS model determines which resource is selected based on non-commensurate and conflicting QoS criteria. Resource selection is easily formulated as a MADM problem which can be solved by many methods. No matter which method is used to solve MADM problem, the weights reflecting the relative importance for the criteria should be determined first. According to the approaches to determining weights, we classify quality-driven scheduling algorithm as lexicographic and Euclidean Distance algorithm. We also employ dynamic strategy to implement load balancing in algorithm which selects the lower load resource, ensuring the high efficiency execution of jobs based on load balancing of the whole grid. The performance evaluation of the algorithm is presented in the next section.

4.1 Lexicographic Algorithm

This algorithm attempts to process jobs as more as possible in terms of preference attribute given by user within the deadline and budget. However, a user may not indicate exactly his/her weights values of attributes, but easily represent his/her preference relation on weights in real world.

Account to the preference relation on weights given by users, we employ a ratio scale to elicit pair-wise comparisons of QoS criteria and an additive aggregation to global weights with Analytic Hierarchy Process (AHP) [9]. Furthermore, to release the strict weight relationship, the equivalent relation among the weights of attributes is tolerable. That is, the revised weight relationship, $w_1 >> w_2 = w_3 >> \ldots >> w_{m-1} = w_m$, is now permissible in lexicographic algorithm. The relaxation seems more flexible and practicable for solving the real-grid problems.

Our approach can be described as follows ($1 \le i \le n$, $1 \le j \le m$, $i \ne j$):

Step 1. Determine the relative weight for these attributes (or criteria) with AHP method and rank the attributes.

Step 2. The attribute with the highest weight is first selected for comparison. Since the normalized decision matrix N has been built as above. Each attribute value has been converted into a numerical value in a range from 0 to 1. The priority of resources is then determined by their relevant values p_{ij} in Matrix N.

Step 3. If some resource attributes possess the same weight, such as $w_i = w_j$, combine these attributes for each resource based on ideal-point approach which is introduced in the next section and go to Step 2.

Step 4. If there are several resources tied with the same attribute value, get the attribute (or attributes) with the next highest weight and compare these tied resources by using Steps 2 and 3.

Step 5. Proceed in this manner until the priority of all resources is determined.

Step 6. Assign the job according to the priority of resources whose current time and processing expenses are within the deadline and budget limits, and remove it from the Unassigned-Jobs-List as long as there exists unassigned jobs.

Lexicographic algorithm aims at assigning the jobs to the best resource satisfying user's preference requirement, thus user could be self-determinated to adjust weights of preference attribute.

4.2 Euclidean Distance Algorithm

In the context of resource selection, the effect of each attribute cannot be considered alone and always be seen as a trade-off with respect to other attributes. Any changes in, for instance, load, credit evaluation, delay and the reliability may change the resource priorities. In light of this, the ideal-point-based approach used in Euclidean Distance algorithm similar as TOPSIS [10] in MADM, seems a suitable method for selection problems since it allows explicit trade-offs and interactions among attributes.

The approach is based on the idea that the best alternative should have the shortest distance from an ideal resource R^*, which is composed of all the best attribute values achievable, while the worst resource R^0 is composed of all the worst attribute values achievable. Then, the goal is to propose a resource R that has the shortest distance D^* from the ideal resource and the farthest distance D^0 from a worst resource simultaneously in the Euclidean space. This algorithm applies the concept of Euclidean space (from a geometrical point of view) to describe the distance D.

$$D = \sqrt{\sum_{s=1}^{n} (x_s - x^*)^2} \tag{4}$$

where x* is an ideal attribute value

To avoid the situation that the candidate resource which has the minimum Euclidean distance from resource R^* may also have a short distance from the resource R^0 as compared to other alternatives, we employ the city block distance [11] expression as follows to solve the contradiction.

$$D = \sum_{s=1}^{n} \left| x_s - x^* \right| \tag{5}$$

Euclidean Distance algorithm selects the candidate resource by similarity to ideal one without any user's preference; consequently, the weights are determined by solving MQoS model automatically. The entropy method [8] is used for assessing the weight in a given problem because, with this method, the decision matrix for a set of candidate resources contains a certain amount of information. Since there is, in selection problems, direct access to the values of the decision matrix, the entropy method is the appropriate evaluation about weights method.

Similarly, we consider the current load as an evaluation for resource that the lowest load of resource should be scheduled first. The scheduling algorithm is described as follows:

Step 1. Normalize the decision matrix N as follows:

$$p_{ij} = \frac{n_{ij}}{\sqrt{\sum_{i=1}^{n} n_{ij}^2}}, \quad j=1, 2, \ldots, m;\ i=1, 2, \ldots, n \tag{6}$$

Step 2. Multiply the columns of the normalized decision matrix P= $\{p_{ij}\}$ by the associated weights. The weighted and normalized decision matrix **V**= $\{v_{ij}\}$is obtained as:

$$v_{ij} = p_{ij} \times w_j \tag{7}$$

where w_j represents the weight of the j^{th} QoS attribute(or criteria).

Step 3. Determine the ideal and worst resources. The ideal and the worst value X_j^*, X_j^0 are determined as the QoS benefit and cost criteria have been determined in matrix N, respectively, as follows ($1 \le i \le n$, $1 \le j \le m$):

$$X_j^* = \max_i v_{ij} \text{, and } X_j^0 = \min_i v_{ij} \tag{8}$$

Step 4. Measure distances from the ideal and worst resources with formula (1). The two distances D_i^*, D_i^0, for each resource, calculated as:

$$D_i^* = \sum_{j=1}^{n} \left| v_{ij} - X_j^* \right| \text{, and } D_i^0 = \sum_{j=1}^{n} \left| v_{ij} - X_j^0 \right| \tag{9}$$

Step 5. Define load factor L_i for resource R_i with the load index ratio a_p (i.e. queue length, memory utilization, I/O operation) and normalized weights λ_p (λ_p <1) given by system as follows:

$$L_i = \sum_{p=1}^{N} \lambda_p \times a_p \text{ ; } i = 1, 2, \ldots, n \tag{10}$$

where $0 \le a_p \le 1$, a_p describes the current load information of nodes gathered from resource monitor, N is the number of a_p, and L_i denotes a relative value of load, that is the bigger the load factor, the bigger the load capacity of resource.

Step 6. Calculate relative closeness of resource R_i to the ideal resource. The relative closeness to the ideal resource can be defined as:

$$C_i = D_i^0 \Big/ (D_i^0 + D_i^*) \quad i = 1, 2, \ldots, n, 0 \le C_i \le 1 \tag{11}$$

It means the larger the closeness the higher the rank.

Step 7. Determine synthetic evaluation T_i as:

$$T_i = (L_i / C_i) \tag{12}$$

Step 8. Assign the jobs according to the ascending order of T_i for resources whose current time and processing expenses are within the deadline and budget limits, and remove it from the Unassigned-Jobs-List as long as there exists unassigned jobs.

5 Performance Evaluation

We modeled and simulated a number of time- and space-shared resources with different characteristics, configurations, and capabilities as those in center of high performance computing of NPU testbed. Table 2 shows the resources simulated using GridSim. We also have modeled a task farming application that consists of 500 jobs.

Table 2: Resources simulated using GridSim

	Resource Type	Nodes, CPU	Cost (G$/CPUsec)
R1	HP RX2600	42, 2	8
R2	Lenovo cluster	18, 2	5
R3	HP SMP	1, 32	3
R4	IBM WorkStation	16, 1	4
R5	SUN WorkStation	8, 1	4

In order to be more generic and precise, we provide our definitions of four criteria: (QoS_1) credit evaluation; (QoS_2) execution cost; (QoS_3) execution time; (QoS_4) reliability, a qualitative attribute which is the degree that a resource will satisfy its users when users demand it. Table 3 shows the four selection criteria and the corresponding assessment data for 5 resources on which there are 100 jobs. Suppose the distinct evaluation grades are defined as H={poor, low, average, good, excellent}, and $P\{H\}=[-1,-0.4,0,0.4,1]^T$ defined by (2). In Table 3, for example, a user may only be able to state that he is 20% percent sure that the reliability is good and 50% percent sure that it is average, expressed G (0.2) and A (0.5) for R_3.

We can obtain the matrix N by using formulas (1) (2):

Table 3:

	QoS^1	QoS^2(G$)	QoS^3(s)	QoS^4
R1	0.95	8	6.25	E(0.7) G(0.1)
R2	0.90	5	11.5	G(0.4)
R3	0.85	3	12	G(0.2) A(0.5)
R4	0.88	4	12.8	A(0.2)
R5	0.92	4	9	G(0.8)

Quantize the qualitative attribute QoS_4 and normalize the Decision Matrix N:

$$N = \begin{vmatrix} 1 & 0 & 1 & 0.148 \\ 0.5 & 0.6 & 0.198 & 0.032 \\ 0 & 1 & 0.122 & 0.016 \\ 0.3 & 0.8 & 0 & 0 \\ 0.7 & 0.8 & 0.580 & 0.064 \end{vmatrix}$$

Suppose that a user gives his the pair-wise comparison matrix Q on the criteria, the cases of two algorithms are addressed as follows:

$$Q = \begin{bmatrix} 1 & 1/3 & 1/4 & 1/4 \\ 3 & 1 & 2 & 3 \\ 4 & 1/2 & 1 & 4/5 \\ 4 & 1/3 & 5/4 & 1 \end{bmatrix}$$

Case 1. We can obtain the weights w = $(0.08, 0.48, 0.24, 0.2)^T$ according to matrix Q by AHP, and the preference order of five resources is given by using lexicographic algorithm, i.e. $R_3 > R_4 > R_5 > R_2 > R_1$.

Case 2. In Euclidean Distance algorithm, the weights w= $(0.1118, 0.019, 0.4208, 0.4480)^T$ are assigned according to the matrix N by entropy method and resources are sorted as R1>R5>R2>R3>R4. We can observe that Euclidean Distance algorithm can optimize the performance of resource selection, and assign the jobs to the best resource as soon as possible.

In addition, we assigned the same computational resources and tasks form different scheduling algorithm such as Euclidean Distance scheduling algorithm under cost and time constraint, time optimization, cost optimization scheduling algorithm.

The CPU utilization comparison is shown in Figure 2. It can be observed that though the utilization of resource R_1 decreased in Euclidean Distance algorithm, the other resources increased at a larger extent. From the view point of the whole system, the algorithm maintains the load-balance of system with dynamic load strategy to adjust the load-balance automatically.

The Average Response Time (ART) is showed respectively in Figure 3. We can conclude Euclidean Distance algorithm obviously surpasses the other two algorithms in ART. This conforms to the market rule completely and indicates the algorithm is effective in actual Grid environment.

The variety of credit criteria referred in the algorithm has been shown in Figure 4. It is indirect to describe the dynamicity of resources, such as entering (R3, R4), and exiting randomly. If there are some unexpected reasons, such as over loading, job suspended, resulting in the increase of execution time, credit evaluation values will decrease correspondingly to reflect dynamic resource scheduling mechanism that makes trades-off between loads and benefit from resources.

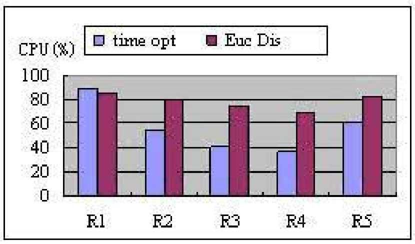

Figure 2: The CPU utilization

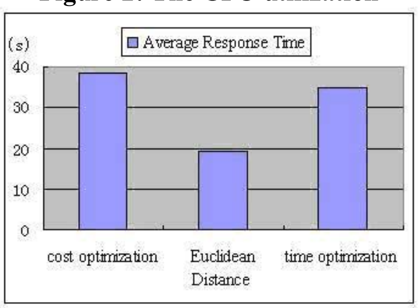

Figure 3: The Average Response Time

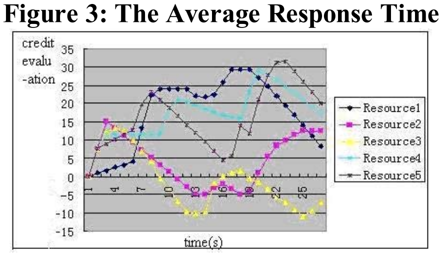

Figure 4: The dynamic state of credit evaluation

6 Conclusions and Future Work

This paper presents a scheduling system with MQoS selecting strategy based on market model, and designs quality-driven scheduling algorithm. It can accomplish the dynamic and autonomy of grid resource more profitably by defining the multiple-dimension QoS criteria of resources for evaluating resource services. Meanwhile, the algorithm adopts the strategies of sorting the QoS preference and the ideal-point-based approach with AHP and entropy respectively for computing weights. We have also conducted experiments to validate the availability of the scheduling algorithm. Results showed that our scheduling algorithm retains the advantages of DBC algorithm and balances the load very well, and it is efficient in terms of resources dynamic and users' QoS requirements.

However, there is still some room for further investigating in our algorithm. We have addressed only the predefined method to identify the static weights which is not in favor of self-adjusting weights. Consequently, the mechanism supporting self-adapting regulation is still a topic of research. In addition, how to choose scheduling frequency is another optimization problem that needs further research.

References

[1] Foster, I., Kesselman, C.. Globus: A Metacomputing Infrastructure Toolkit. International Journal of Supercomputer Applications. 11(2):115-128, 1997.

[2] S. Chapin, J. Karpovich, A. Grimshaw. The Legion Resource Management System. In：Proceedings of the 5th Workshop on Job Scheduling Strategies for Parallel Processing. 2001. USA: IEEE Computer Society Press, 1999. 69~76,129~137.

[3] Maheswaran, M. Ali, S. Siegal. Dynamic matching and scheduling of a class of independent tasks onto heterogeneous computing systems. HCW'99, 1999, 30-44.

[4] O.H. Ibarra and C. E. Kim. Heuristic algorithms for scheduling independent tasks on nonidentical processors. Journal of the ACM, 24(2), April 1977, 280-289.

[5] R. Buyya, J. Giddy. An evaluation of economy-based resource trading and scheduling on computational power Grids for parameter sweep applications. Proceedings of the 2nd International Workshop on Active Middleware Services (AMS 2000), Pittsburgh, PA, 1 August 2000.

[6] R.Buyya, D.Abramson. Economic Models for Resource Management and Scheduling in Grid Computing. The Journal of Concurrency and Computation: Practice and Experience（CCPE）, Wiley Press. May 2002.

[7] YANG JB, SEN P. A general multi-level evaluation process for hybrid MADM with uncertainty. [J]. IEEE Transactions on Systems, Man, and Cybernetics, 1994, 24(10): 1458-1473.

[8] E. S. Soofi. Generalized entropy based weights for multi-attribute value models. Oper. Res., vol. 38, pp. 362－363, 1990.

[9] Saty, T.L.. The Analytic Hierarchy Process. McGraw Hill Inc. [J]. NY, 1980.

[10] Hwang, Ching - Lai & Yoon, Kwangsun. Multiple attribute decision making. Birlin: Springer -Verlag, 1981.

[11] B.V. Dasarathy, SMART: similarity measure anchored ranking techniquefor the analysis of multi-dimensional data analysis, IEEE Trans. Syst. Man Cybern. SMC-6 10 (1976) 708–711.

0-7695-2934-8/07 $25.00 © 2007 IEEE