Hierarchical Structure Resource Scheduling Model Based on Grid

WANG Na[1]
Zheng Zhou Information Science and Technology Institute
Zheng Zhou, China

WEI Bo[2]
Zheng Zhou Information Science and Technology Institute
Zheng Zhou, China
Weibo7516@sina.com

ZHANG Heng-wei[3]
Zheng Zhou Information Science and Technology Institute
Zheng Zhou, China

WANG Jin-dong [4]
Zheng Zhou Information Science and Technology Institute
Zheng Zhou, China

***Abstract*—Resource matching is the core function of the grid. This paper analyzed the characteristics of the grid, designed a Hierarchical Resource Scheduling Model for data resources in grid and a new resource scheduling algorithm. This model can combine each function units of resource scheduling organically to meet with resource scheduling requirements of different goals, different levels, and different particle size involved in grid resource scheduling. Finally, a new resource scheduling algorithm was designed based on the resource object space distance. The experimental results show that the algorithm can effectively balance the performance and cost index in less computation, realize resource scheduling, and won't cause too much burden to the system.**

Keywords-grid; resource scheduling model; resource scheduling algorithm

I. INTRODUCTION

In recent years, with the rapid development of Internet technology and applications, grid technology is a research focus in the field of distributed systems. Grid is a large-scale integration of resources and computing environment, it is able to combine the distributed computing and data resources in different management organizations to form a virtual organization [1]. To provide users with high-performance computing services and data storage, transmission service [2]. Grid has the following characteristics: (1) Grid resources are wide distributed, between resources, resources and users often connected through wide area network; (2) The state of resources and the grid is changed dynamically; (3) As a public infrastructure, the number of grid resources is huge; (4) Grid resources can join or leave the grid system at any time. So we need an effective resource management framework to manage the grid resources.

In the existing grid system, there are three resource scheduling model: centralized, distributed and hierarchical manner. Because centralized resource scheduling model have problems of center node performance bottleneck and single node failure, distributed resource scheduling model have problems of resource node security risks and lower resource discovery performance. As the combination of both, hierarchical resource scheduling can solve the above problems effectively. So, hierarchical resource scheduling is adopted by most of resource scheduling systems. For characteristics of the grid and shortages of existing resource scheduling model, this paper designed a hierarchical resource scheduling model. Resource scheduling domain model, scheduling technology model, and scheduling algorithm had been designed in detail.

II. HIERARCHICAL RESOURCE SCHEDULING MODEL

A. Hierarchical Structure

Based on requirements of hierarchic structure and management of each layer domain in grid resource scheduling, this paper designed a hierarchical resource scheduling model. Each layer is divided into several logic resource scheduling domains, Such as Figure 1 shown.

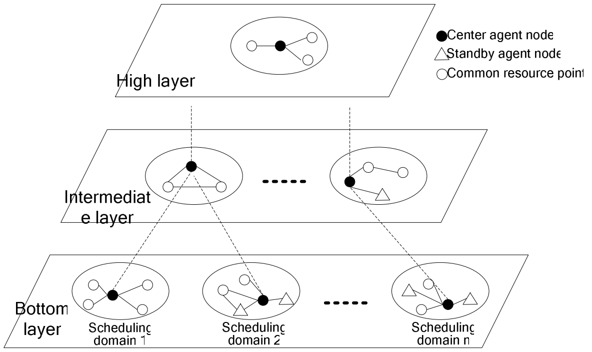

Figure 1. Hierarchical resource scheduling model

- Resource scheduling model designed to hierarchical structure, the priority of higher layer is higher than lower layer, the bottom resource scheduling center can provide services for higher layer resource scheduling center, higher resource scheduling center can coordinate bottom resource scheduling center to achieve the prearranged functions, so as to reduce system overhead;

Fund from Henan Science and Technology Agency Num: 082102320010

- Each layer was divided into several logic resource scheduling domains, resource nodes in scheduling domain were divided into center agent node, standby agent nodes, and common resource nodes;
- Center agent node is borne by the node with best performance in scheduling domain, take charge of resource scheduling operation of this domain. In resource scheduling model, resource scheduling center is located at the center agent node. The resource scheduling center of each scheduling domain connects with higher layer resource scheduling center. Resource scheduling among bottom scheduling domains can be achieved through their higher resource scheduling centers;
- Standby agent node is borne by the node whose performance after center agent node, and carry out resource scheduling operation when center agent node have a failure to prevent resource scheduling service interruption;
- Common resource node can be resource provider or resource request.

B. Resource Scheduling Domain Model

Scheduling domain interior and connection type among domains in hierarchical resource scheduling framework were designed in detail. This model could effectively improve the efficiency of resource scheduling.

For the above problems, this paper designed a new resource scheduling domain model based on resource scheduling framework. That is, p2p network [3] was used in the internal layer of resource scheduling domain and introduced the concept of center proxy node. The center agent node is responsible for resource scheduling tasks in this domain, outer layer used connection type based on WS. The resource scheduling in lower level management domains can be done through the center agent node in higher level management domain. And we used WS technology to connect higher and lower layer scheduling domains. The model was shown in Figure 2.

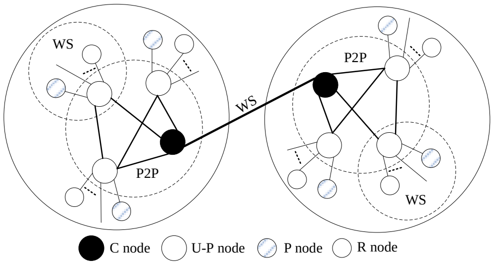

Figure 2. Resources Scheduling Domain Model

- Based on WS technology [4], the outer layer of scheduling domain in this resource scheduling model resolved problems of resource node security risk and resource discovery inefficiency in distributed resource scheduling. Internal layer uses p2p technology to resolve the problem of center node performance bottleneck and single node failure in centralized resource scheduling;
- The C (center) is the center agent node, it's also a U-P node; the U-P node is the resource register center in the outer WS structure, it's also a peer node in the P2P structure; The P (provide) is a resource provide node, constitute resource providers of WS structure, and resource providers register with resource scheduling center through this node; The R (require) node is a resource request node, constitute resource requests of WS structure, and resource requestors request resources from resource scheduling center through this node.

C. Resource Scheduling Technology Model

Resource scheduling center located in the center agent node of scheduling domain. It's the center of resource scheduling in the domain.

Resource scheduling center is responsible for the creation of the work domains, the collection of resource information, and receiving resource request submitted from the resource requestor interface, and take corresponding operation on the request. The users simply need to submit resource requests to the resource requestor interface, the specific number of the resource providers, as well as how the request schedule resources is transparent to the users. The resource scheduling technology model designed in this paper shown as Figure 3.

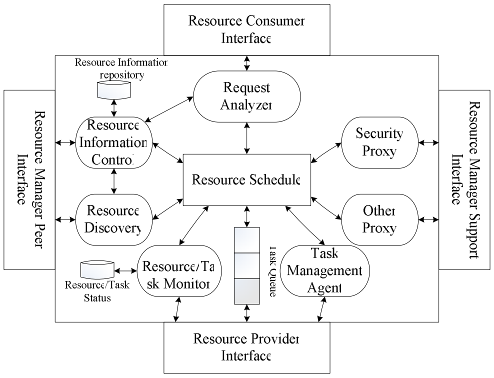
Resource Consumer Interface
Resource Information repository
Request Analyzer
Resource Information Control
Security Proxy
Resource Manager Peer Interface
Resource Schedule
Resource Manager Support Interface
Resource Discovery
Other Proxy
Task Queue
Resource/Task Status
Resource/Task Monitor
Task Management Agent
Resource Provider Interface

Figure 3. Resource scheduling technology model

Resource scheduling technology model consists of four interfaces and several component units, and the specific name shown as figure. Resource scheduling is the core of scheduling model. The main functions of four interfaces and several component units are as follows:

- Resource scheduling: resource scheduling unit is the core of the whole resource scheduling model, it is the bond between resource requestors and providers. Based on resource request parameter, this unit matched the optimal resource from the available resource list to meet parameter requirements, returned the unique identifier of this resource to requestors and will transmit "task-resource" matching relationship to task queue unit;

- Resource requestor interface: resource requestor request resources from resource scheduling center through resource requestor interface;

  - Request analysis: this unit will preprocess user request parameters to create resource scheduling parameters.

- Resource scheduling equivalence interface;

  - Resource filter: it is responsible for filtering high risk, low credible and real-time heavy load resource to ensure the service quality of resources provided to resource list in resource scheduling unit;
  - Resource register: only registered resources can be scheduled and used by users as legal resource in ISDS;
  - Resource discovery: it links resource unacquainted resource in a grid environment with user requests.

- Resource provider interface: resource provider provide service to higher-layer application through resource provider interface;

  - Resource monitoring: it monitors status information to find abnormity, and feed back abnormal information to resource scheduling unit;
  - Resource queen: it is responsible for the maintenance of job queue;
  - Resource migration: it provides resource migration service that moves resource used by several users to the lower network load node to reduce system consumption.

- Resource scheduling support interface: it provides support function needed by resource scheduling cent.

  - Security agent: this unit is responsible for providing corresponding security to resource scheduling system;
  - Other agent： other agent services needed by system can be designed in this unit.

III. Resource Scheduling Algorithm

A. Related Definitions

1) Definition 1: the performance vector and the cost vector

Resource request parameters can be divided into performance parameters and cost parameters, we use $(\varepsilon_1,\varepsilon_2,\cdots,\varepsilon_m)$ and $(\eta_1,\eta_2,\cdots,\eta_n)$ to indicate. Among these, $\varepsilon_i\left(1\le i\le m\right)$ and $\eta_j\left(1\le j\le n\right)$ respectively indicate variable parameters of a particular aspect on performance and cost of the described resources in the current state.

The variable parameters in these two vectors represent the actual values of the performance and cost. But the dimension and magnitude of each parameter are different. In order to prevent the neglect of small magnitude variable parameters, we use the treatment of non-dimension method to deal with the raw data. In this paper, we use linear method, as follows.

$$y_i=\frac{x_i}{x_{i\max}}$$

After the performance parameters and the cost parameters were normalized by the treatment of non-dimension method we get the performance vector $P(p_1,p_2,\cdots,p_m)$ and the cost vector $C(c_1,c_2,\cdots,c_n)$.

2) Definition 2: the cost-performance correlation matrix

$$CP=\begin{bmatrix} cp_{11} & cp_{12} & \cdots & cp_{1m} \\ cp_{21} & cp_{22} & \cdots & cp_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ cp_{n1} & cp_{n2} & \cdots & cp_{nm} \end{bmatrix}=(cp_{ij})_{n\times m}$$

In the matrix, $cp_{ij},(0\le cp_{ij}\le 1,\sum_{j=1}^{m}cp_{ij}=1)$ represents relativity of p_j and c_i .If p_j and c_i are completely correlation，then $cp_{ij}=1$.If p_j and c_i are not related to each other, then $cp_{ij}=0$ .Under other conditions the value would be between 0 and 1.

3) Definition 3: the cost-performance vector of resource

Firstly, we calculate the vector $(\frac{p_1}{c_1}\cdot cp_{11},\cdots,\frac{p_m}{c_1}\cdot cp_{1m},\cdots\cdots,\frac{p_1}{c_n}\cdot cp_{n1},\cdots,\frac{p_m}{c_n}\cdot cp_{nm})$ .Before calculation we eliminate items in which $cp_{ij}=0$ or $c_i=0$, then we get the cost-performance vector $R(r_1,r_2,\cdots,r_l)$.

4) Definition 4: the cost-performance vector matrix of existing resources

$$H_t=(h_{1t},h_{2t},\cdots,h_{kt})=\begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1k} \\ h_{21} & h_{22} & \cdots & h_{2k} \\ \vdots & \vdots & & \vdots \\ h_{l1} & h_{l2} & \cdots & h_{lk} \end{bmatrix}=\left(h_{ij}\right)_{l\times k}$$

h_{it} is the cost-performance vector of each existing resource. k is the number of the existing resources. t is the time we calculate these cost-performance vectors.

5) Definition 5: the membership vector

The membership vector μ_t consisted of values of how the of requested resource attached to the existing resources' cost-performance vectors at time t . μ_t is expressed as:

$$\mu_t=\left(\mu_{1t},\mu_{2t},\cdots,\mu_{kt}\right)^T\ (0\le\mu_{it}\le 1,\sum_{i=1}^{k}\mu_{it}=1)$$

B. Resource Matching Method

In the resource matching process, we introduce the concept of distance space in Functional Analysis. The distance space is the extension of n-dimensional Eculid space, which summed up the most essential things in the concept of distance between two nodes in the Eculid space. As a result, it established abstract distance space theory based on the general set. According to the definition of the distance space we can map the cost-performance vectors of the existing k Kinds of resources to k basis nodes of multi-dimension space. The cost-performance vector of the requested resource can also have mapped nodes in the multi-dimension space. Through the concept of distance space we can calculate the distance between the node and the other nodes, and in this way we can find the most suitable resource out for the resource request.

Assume that at t moment the cost-performance vector matrix of existing resources is $H_t = (h_{1t}, h_{2t}, \cdots, h_{kt})$, then the method of determining which resource best meets the resource request is as follows:

First, we handle the performance parameters $(\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_m)$ and the cost parameters $(\eta_1, \eta_2, \cdots, \eta_n)$ of requested resource with the treatment of non-dimension method. Then according to definition 3, we use the cost-performance correlation matrix to calculate the cost-performance vector of resource request:

$$x = (x_1, x_2, \cdots, x_l)^T \tag{1}$$

Through the H_t in Definition 4, we can get the cost-performance vector of resource j:

$$h_{jt} = (h_{1j}, h_{2j}, \cdots, h_{lj})^T \tag{2}$$

Because $\forall x = (x_1, x_2, \cdots, x_l)^T$, $h_{jt} = (h_{1j}, h_{2j}, \cdots, h_{lj})^T \in R^l$. So, according to the definition of the distance space, the distance between $x = (x_1, x_2, \cdots, x_l)^T$ and $h_{jt} = (h_{1j}, h_{2j}, \cdots, h_{lj})^T$ can be expressed as:

$$d_p(x, h_{jt}) = \left(\sum_{e=1}^{l} \left| x_e - h_{ej} \right|^p \right)^{\frac{1}{p}}, 1 \le p < \infty \tag{3}$$

In the n-dimensional Euclid space, when p = 2, we call it Euclidean distance:

$$d_2(x, h_{jt}) = \left(\sum_{e=1}^{l} \left| x_e - h_{ej} \right|^2 \right)^{\frac{1}{2}} \tag{4}$$

In order to have a more complete description between the difference of x and h_{jt} , we introduce the concept of membership and weight coefficient. Then we get a weighted Euclidean distance d_{jt} :

$$d_{jt} = \mu_{jt} \left(\sum_{e=1}^{l} w_e \left| x_e - h_{ej} \right|^2 \right)^{\frac{1}{2}} \tag{5}$$

Each w_i in $W = (w_1, w_2, \cdots, w_l)$ is a weight coefficient corresponding to a parameter variable in the cost-performance vector.

According to equation (5), we establish the objective function to calculate the state membership vector $\mu_t = (\mu_{1t}, \mu_{2t}, \cdots, \mu_{kt})^T$. The significance of the objective function is the equation $\sum_{j=1}^{k} (d_{jt})^2$ is the least.

$$\min[f(\mu_{jt})] = \min \sum_{j=1}^{k} (d_{jt})^2 = \min \sum_{j=1}^{k} \mu_{jt}^2 \left[\sum_{e=1}^{l} w_e \left| x_e - h_{ej} \right|^2 \right] \tag{6}$$

According to equation (5), we construct Lagrangian function as follows:

$$L(\mu_{jt}, \lambda) = \sum_{j=1}^{k} \mu_{jt}^2 \left[\sum_{e=1}^{l} w_e \left| x_e - h_{ej} \right|^2 \right] - \lambda \left(\sum_{j=1}^{k} \mu_{jt} - 1 \right) \tag{7}$$

$$\partial L(\mu_{jt}, \lambda) / \partial \mu_{jt} = 2\mu_{jt} \left[\sum_{e=1}^{l} w_e \left| x_e - h_{ej} \right|^2 \right] - \lambda = 0 \tag{8}$$

$$\mu_{jt} = \lambda \Big/ \left\{ 2 \left[\sum_{e=1}^{l} w_e \left| x_e - h_{ej} \right|^2 \right] \right\} \tag{9}$$

$$\partial L(\mu_{jt}, \lambda) / \partial \lambda = 1 - \sum_{j=1}^{k} \mu_{jt} = 0 \tag{10}$$

According to equation (9) and (10), we obtain:

$$\lambda = 2 \Big/ \left\{ \sum_{j=1}^{k} 1 \Big/ \left[\sum_{e=1}^{l} w_e \left| x_e - h_{ej} \right|^2 \right] \right\} \tag{11}$$

According to equation (9) and (11), we obtain:

$$\mu_{jt} = \frac{1 \Big/ \left[\sum_{e=1}^{l} w_e \left| x_e - h_{ej} \right|^2 \right]}{\sum_{j=1}^{k} \left\{ 1 \Big/ \left[\sum_{e=1}^{l} w_e \left| x_e - h_{ej} \right|^2 \right] \right\}} \tag{12}$$

We can know from (12), when $x = (x_1, x_2, \cdots, x_l)^T$ and $h_{jt} = (h_{1j}, h_{2j}, \cdots, h_{lj})^T$ are same, the denominator is 0. So we make $\mu_{jt} = 1$ at this time.

As a result, we can get the membership vector $\mu_t = (\mu_{1t}, \mu_{2t}, \cdots, \mu_{kt})^T$ of requested resource. Based on the principle of maximum membership degree, the best suitable resource that requested can be found.

C. Algorithm Design

1) Algorithm input:

a) The performance parameters $(\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_m)$ and the cost parameters $(\eta_1, \eta_2, \cdots, \eta_n)$ provided by resource requester.

b) The performance parameters and the cost parameters of all existing resources at t moment.

c) The cost-performance correlation matrix:

$$CP = \begin{bmatrix} cp_{11} & cp_{12} & \cdots & cp_{1m} \\ cp_{21} & cp_{22} & \cdots & cp_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ cp_{n1} & cp_{n2} & \cdots & cp_{nm} \end{bmatrix}$$

d) Weight coefficient of cost-performance vector: $W = (w_1, w_2, \cdots, w_l)$.

Among them, the cost-performance correlation matrix and the weight coefficient of cost-performance vector can be set as default value; The performance parameters and the cost parameters of all existing resources only need to refresh again when there are update actions.

2) Algorithm process:

Step1: use the treatment of non-dimension method to deal with the performance parameters $(\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_m)$ and the cost parameters $(\eta_1, \eta_2, \cdots, \eta_n)$ provided by resource requester to get the performance vector $P(p_1, p_2, \cdots, p_m)$ and the cost vector $C(c_1, c_2, \cdots, c_n)$.

Step2: According to the cost-performance correlation matrix CP and definition 3, the cost-performance vector $x = (x_1, x_2, \cdots, x_l)^T$ of requested resource can be calculated.

Step3: According to step 1 and 2, the cost-performance vectors of all existing resources can be calculated. And we can get the existing resources' cost-performance vector matrix H_t in definition 4.

Step4: Based on the resource matching method, the resource requester's membership vector $\mu_t = (\mu_{1t}, \mu_{2t}, \cdots, \mu_{kt})^T$ can calculated.

Step5: Find the $\max \mu_{jt} (1 \le j \le k)$ in the membership vector $\mu_t = (\mu_{1t}, \mu_{2t}, \cdots, \mu_{kt})^T$.

Step6: According to $\max \mu_{jt} (1 \le j \le k)$, the resource most corresponded with the resource requester's parameters is the resource j.

D. Simulation Experiment

In order to prove the effectiveness of the resource matching algorithm, we use Matlab to simulate the algorithm. The steps are as follow:

In this experiment, we use the intrusion detection software resource as an example. According to the characteristics of intrusion detection software, we choose the performance parameters and the cost parameters [5] as TABLE I. Among them, the first 6 groups are the existing intrusion detection software resources, the last 4 groups are the request parameters provided by the resource requester.

TABLE I. THE PERFORMANCE PARAMETERS AND THE COST PARAMETERS

number	performance parameter				cost parameter			
	single/ network	detection speed	monitoring accuracy	monitoring scope	software size	installation costs	memory size occupied	bandwidth size occupied
Ⅰ	1	60	30	3	70	1	76	1
Ⅱ	1	90	55	8	110	4	115	2.5
Ⅲ	1	70	40	6	90	2	95	2
Ⅳ	2	5	20	26	30	7	43	5
Ⅴ	2	7	25	30	40	9	50	6
Ⅵ	2	9	35	45	55	12	66	8
1	1	76	50	7	100	2.9	104	2.2
2	2	6.5	21	27	36	7.4	45	5.3
3	1	60	42	7	80	3	80	6
4	2	8	30	32	40	12	48	7

According to expert experience, the cost-performance correlation matrix and the weight coefficient of cost-performance vector have been given as follows:

$$CP = \begin{bmatrix} 0 & 0.2 & 0.6 & 0.3 \\ 0.3 & 0 & 0 & 0 \\ 0 & 0 & 0.4 & 0 \\ 0.7 & 0.8 & 0 & 0.7 \end{bmatrix}$$

$$W = (0.05 \quad 0.12 \quad 0.16 \quad 0.09 \quad 0.17 \quad 0.16 \quad 0.15 \quad 0.1)$$

After the treatment of non-dimension to the existing 6 kinds of existing intrusion detection software's performance parameters and cost parameters, and according to the cost-performance correlation matrix[6], we get the existing resources' cost-performance vector matrix as follows:

$$H_t = \begin{bmatrix} 1.2 & 0.4539 & 1.2 & 0.3143 & 0.0733 & 0.3301 & 0.64 & 0.3733 \\ 0.3 & 0.3 & 0.48 & 0.3 & 0.1244 & 0.4 & 0.4267 & 0.3982 \\ 0.6 & 0.3632 & 0.6 & 0.2852 & 0.1141 & 0.3522 & 0.64 & 0.3733 \\ 0.3429 & 1.6047 & 0.48 & 0.0611 & 1.483 & 0.389 & 0.7924 & 0.6471 \\ 0.2667 & 1.38 & 0.4 & 0.0642 & 1.2833 & 0.4182 & 0.7111 & 0.6222 \\ 0.2 & 1.0455 & 0.3 & 0.06 & 1.4 & 0.4435 & 0.8 & 0.7 \end{bmatrix}^T$$

Then calculate the requester's cost-performance vector separately. According to the existing resources' cost-performance vector matrix, we can get the membership matrix of the requesters as follows:

$$U = \begin{bmatrix} 0.0734 & 0.4597 & 0.4285 & 0.0134 & 0.0137 & 0.0114 \\ 0.0044 & 0.0024 & 0.0036 & 0.8243 & 0.1536 & 0.0118 \\ 0.231 & 0.1786 & 0.388 & 0.073 & 0.0745 & 0.0548 \\ 0.0199 & 0.0153 & 0.019 & 0.1404 & 0.4574 & 0.348 \end{bmatrix}^T$$

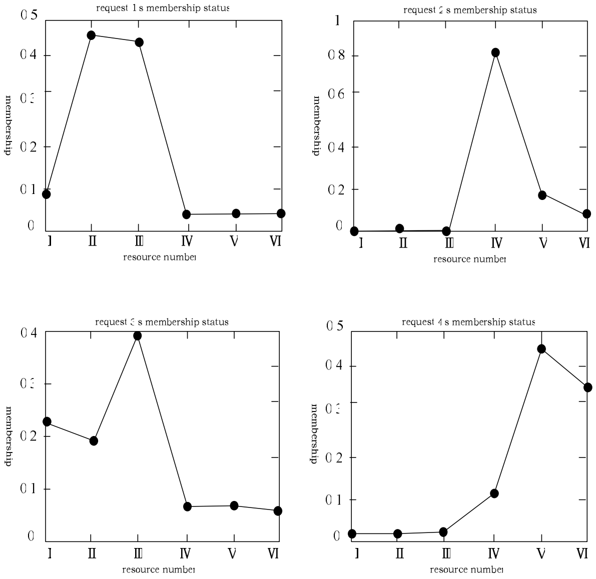

Figure 4. View of the resource requests' membership status

As shown in the figure, the resource numbers requested by the 4 requesters are resource Ⅱ, Ⅳ, Ⅲ and Ⅴ. The resource Ⅳ suits the request 2 best. The results show that the algorithm suits the actual situation in the network environment.

IV. CONCLUSIONS

This paper designed a Hierarchical Resource Scheduling Model. Hierarchical structure, scheduling domain model, scheduling technology model were designed and explained in detail. This model can resolved problems of center node performance bottleneck and single node failure in centralized resource scheduling, also problems of resource node security risk and resource discovery inefficiency in distributed resource scheduling effectively. Besides, this paper designed a new resource scheduling algorithm based on weighted Euclidean distance. From the simulation results, we knew that the algorithm can effectively achieve the resource matching in a smaller amount of computation, and won't cause extreme burden to resource management system. The algorithm constrained by both performance and cost.

REFERENCES

[1] Klaus Krauter, Rajkumar Buyya, Muthucumaru Maheswaran, "A taxonomy and survey of grid resource management systems for distributed computing," SOFTWARE—PRACTICE AND EXPERIENCE, vol.32, pp.135-164, 2002.

[2] Ling Jianning, Zong Shiqiang, "Research on Resource Management Frame and Algorithm of Simulation Grid", Computer and modernization, 2010, No.7, pp.68-71.

[3] Yanqiu Meng, Zhongjie Wang, Xiaofei Xu, Xianzhi Wang, "A Generalized Service Resource Management Framework", 2010 International Conference on Service Sciences, 2010, pp.329-334.

[4] ZHONG Feng, "Research on Coal Enterprise Grid Resource Management Model", 2010 International Conference on Computer and Computational Intelligence (ICCCI 2010), 2010, pp.v2-394-398.

[5] Tian Yilin, Zhao Yinde. "Resourch on grid management of public crisis resources", International Conference on Computer Science and Service System (CSSS), 2011. pp.642-645.

[6] Parmer G, West R. HiRes. "A System for Predictable Hierarchical Resource Management", 2011 17th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), 2011. pp.180-190.