

MRMT: An Algorithm of Matching Resource for Multi-Tasks

Min Yan*, Jiantao Zhou*

*College of Computer Science, Inner Mongolia University, Hohhot, China

yanmin0614@163.com, cszjtao@imu.edu.cn

Abstract— While most of work on task-resource matching in centreless network has focus on evaluating multi-QoS constraints in quantity, our work addresses this problem from types of QoS constraints, objectivity of matching process, and concurrency of multiple matches. An algorithm of matching resources for multi-tasks synchronously, MRMT, is presented. Firstly, an improved analytic hierarchy process (AHP) method is introduced. Secondly, the Gale-Shapley principle and methods are applied to optimize synchronous matches. The algorithm MRMT is preventive for accumulation of inconsistency with multi-QoS constraints, is especially effective for selecting optimal resources synchronously, and is available for maximizing efficiency for matched resources as possible.

Keywords— Multi-Tasks, Resource, Concurrency, AHP, Gale-Shapley

I. INTRODUCTION

With the development of information technology, novel infrastructures and applications in modern network are getting more and more complicated. According to delivered services, we label some distributed network environments like cooperation, grid resource sharing, cloud computing, etc., as centreless network in this paper. Centreless network distinct from traditional distributed or cluster network by its focus on point to point interconnection of resources^[1~3]. A resource in centreless network is software that performs some designated functions or hardware that contributes to a sub-network. There are always lots of independent resources in centreless network, while these resources always interconnect with each other in point to point freely. If some resources broke down, the centreless network would be designed to maintain performance. In centreless network, different resources make themselves available to different tasks in different degrees for different QoS constraints. Meanwhile different tasks also have different requirements in both types and qualities for all kinds of resources. While the same type of resources and tasks, however, paired results between them should not be varied with different matching sequences.

Due to issues such as complexity, dynamicity, autonomy, etc., resource matching in centreless network, for multi-tasks synchronously, is more challenging than in traditional distributed or cluster network environments. In order to select optimal resources for all multi-tasks concurrently, an algorithm of matching resources for multi-tasks (MRMT) is present in

this paper. Firstly, Analytic Hierarchy Process (AHP) method is used in MRMT to give ordered vectors of candidates to tasks/resources. In order to evaluate QoS constraints more objectively, we improve traditional AHP method by build a recursive inter-level sorting process with consistency test^[4, 5]. Secondly, the Gale-Shapley principle and methods are applied in MRMT to build matching relations between resources and tasks synchronously. And it is more appropriate than usual weighted Hungarian method at complexity.

The remainder is organized as follows. In section II, related work is summarized and compared. Then in section III and IV, process and key points of MRMT are given respectively. Finally, concluding remarks are given in section V.

II. RELATED WORK

It is one of original intents of task-resource matching, in centreless network, to run as many tasks as possible synchronously in a certain period time. Besides the time factor, more QoS constraints in quantity and types should be evaluated in current matching algorithms. Our work evaluates multiple QoS constraints more fully and objectively in MRMT by using improved AHP, and works out optimal paired results between resources and tasks synchronously.

First Come First Served (FCFS) and Round Robin (RR) are typical matching algorithms in traditional network. They are easy to implement, but are inefficient in matching performance. While we match resources with FCFS or RR, it is frequent that some tasks are suspended meanwhile some resources are in ready status.

Evaluating both factors of time and load, some heuristic methods such as Min-Min, Min-Max, and Genetic Algorithm were studied^[6~8]. But other QoS constraints were not mentioned at all. Another QoS constraint bandwidth was evaluated in QoS Guided Min-Min algorithm, while more constraints should be introduced in its future work^[9]. More QoS constraints in quantity were evaluated in some improved QoS guided Min-Min algorithms by redesigning benefit functions and linear models^[10~18]. But dependent QoS constraints and their relations could not be evaluated in these improved algorithms. These improved algorithms were also difficult to make themselves suitable for dynamic and heterogeneous resources in centreless network.

In a word, existing matching algorithms were short in evaluating QoS constraints more or less in quantity and types,

and did nothing about matching resources for multi-tasks synchronously. With the algorithm of MRMT, QoS constraints in matching will be evaluated more objectively in quantity and more flexibly in types, and optimal resources will be matched out for multiple tasks synchronously.

III. MRMT ALGORITHM

In this section, process and key points of MRMT are introduced.

A. Introduction of MRMT

The process of MRMT can be organized roughly in follow 3 steps, shown as Fig. 1.

- 1) To order candidate resources for multi-tasks respectively with improved AHP, and vice versa.
- 2) To construct a priority matrix of multi-tasks and all candidate resources.
- 3) To confirm paired results between multi-tasks and corresponding resources with Gale-Shapley.

Shown as Fig. 1, some important issues must be introduced here. Firstly, AHP method used our work is improved. In order to get objective priorities of resources to one task (and vice versa), the inter-level sorting of AHP is reorganized into a recursive process, and consistency test is referred into every iterations. Secondly, both priorities of candidates to tasks and resources are necessary in confirming final matching results. If there was the same matched resource for different tasks, we would evaluate constraints of every resource and compute priorities of tasks to the resource. So a priority matrix of matching between multiple tasks and resources is constructed with two-tuples. Finally, Gale-Shapley is used to give match-relations between resources and tasks.

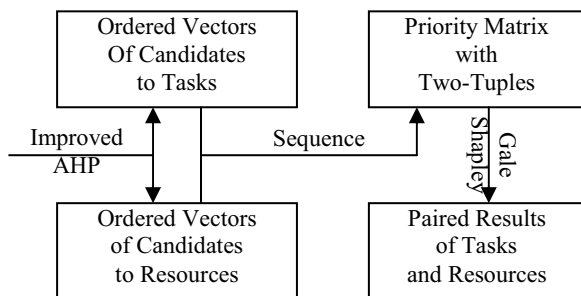


Figure 1. Process of MRMT

B. AHP and Improvement

It does is a novel application of AHP in resource management in centreless network. Based on the principle of similarity theory, multi-QoS constraints and their relations in matches can be formulated as Multiple Attribute Decision Making (MADM) problem. AHP is a typical method for MADM in economic management. To make AHP more efficient and objective in our work, we give evaluating constraints of resources to one task as an example, and summarize process of improved AHP as follow steps.

- 1) To search candidate resources for the tasks with predicting techniques.
- 2) To model QoS constraints and their dependencies in a hierarchy tree.
- 3) To evaluate importance of constraints or resources who depend on the same node.
- 4) To find absolute importance of resources to the task in a vector.

There are two key points in our AHP steps are improved. Firstly, the step 4 is organized into a recursive process. Secondly, inaccuracy in steps is minimized in the recursive process.

We explain steps of improved AHP in detail. To confirm candidate resources that are satisfy all QoS constraints of a task, we mark the task and its k constraints in bottom level as $T=[Q_1, Q_2...Q_k]$, and mark a pre-candidate resource and its n attributes in bottom level as $R=[Q_1, Q_2...Q_n]$. Follow steps decide whether the resource R is an available candidate to the task T .

- 1) $\forall Q_j: Q_j \in T$ where $j \in [1, k]$, jump to step b if $Q_j \in R$, and jump to step d otherwise.
- 2) $\forall Q_j: Q_j \in T$, jump to step c if $|Q_j| \leq |Q_{rj}|$, and jump to step d otherwise, where Q_{rj} is the corresponding QoS constraint of Q_j in the vector R , and $j \in [1, k]$, $rj \in [1, n]$.
- 3) The resource R satisfies all QoS constraints, and set it as one of T 's candidates, over.
- 4) The resource R is not in T 's candidates, over.

After getting candidate resources of a task, we describe relations of QoS constraints/attributes. As most work on task-resources matching has focus on expanding QoS constraints in quantity, we address this problem in types. In order to describe all types of QoS constraints, we label the task in goal level, label constraints in criteria levels, label candidate resources in alternative level, and construct a hierarchy tree with three types of levels, shown as Fig. 2. The task in the goal level depends on several constraints in the first top criteria level in Fig. 2. Refining constraints q_1 into sub-constraints q_{11} and q_{12} , shown as the second top criteria level in Fig. 2, can continue if it is necessary. All nethermost constraints in criteria levels are facets of candidate resources in alternative level in Fig. 2.

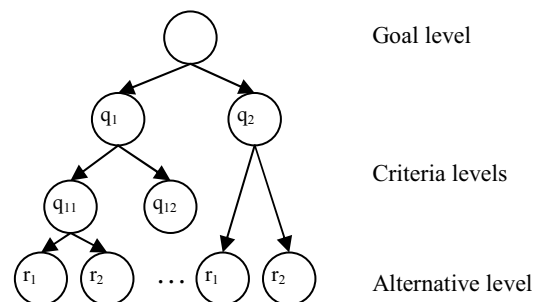


Figure 2. Hierarchy Tree

Labelling nodes dominated by the same point co-nodes, Fig. 2 shows dependency of QoS constraints. In order to evaluate relative importance between co-nodes in a vector, we firstly construct their preference matrix by paired comparisons^[4] and

predicting techniques. If co-node q_{11} depended on constraint q_1 more than q_{12} with triple, the preference matrix of q_{11} and q_{12} could be constructed as $A_1 = \begin{bmatrix} 1 & 3 \\ 1/3 & 1 \end{bmatrix}$. Constructing a preference matrix of n co-nodes, there are $O(n^2)$ comparisons happened. In other words, the algorithm MRMT in our work is flexible in width of hierarchy tree, and co-nodes are limited to nine normally.

A preference matrix gives paired importance between co-nodes. There are lots of ranking methods can be used to transact co-nodes' matrix into a vector^[19]. Eigenvector Method (EM) and Least Deviation Method (LDM) are typical ranking methods of approximate evaluation and optimization separately. Consistency proportion function $CR=CI/RI_n$ is always used in ranking all matrixes to control inaccuracy. As one of distinguish from normal AHP, consistency test also be used in inter-level sorting process in our work.

To find the optimal resource in candidates for a task, we have to integrate co-nodes' relative importance in all levels together, so that final importance of all candidate resources to the task in goal level can be given for evaluating all QoS constraints. We call the process of integration inter-level sorting.

An improved inter-level sorting is provided in MRMT, shown as Fig. 3. In traditional sorting process, all relative importance of co-nodes is integrated directly from top to bottom along real lines in Fig. 4. As a result of accumulation of inaccuracies, however, the final importance of resources is imaginably un-objective. To control accumulation of inaccuracies in all levels in minimum, it is necessary in our work to use consistency function $CR=CI/RI_n$ in all integrations. A flow chart in Fig. 3 gives a novel recursive process of inter-level sorting. It is apparent that the recursive process begins and ends both with root. All nodes that abstracted form constraints and candidates are processed as current nodes separately. Three issues are important in "Process current node" in Fig. 3. Firstly, integration

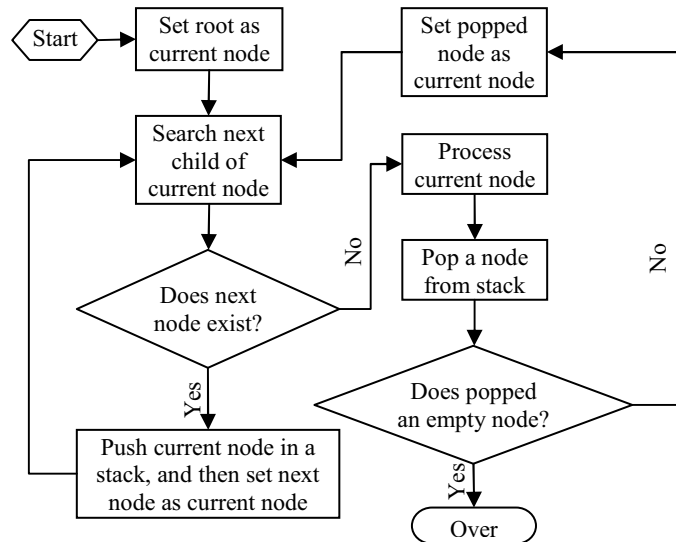


Figure 3. Process of Inter-level Sorting

operations in inheritance nodes and brother nodes are multiplications and additions separately. Secondly, brother nodes from candidates are not integrated together at all. Finally, as an important feature in our work, consistency function $CR=CI/RI_n$ must be embedded in all integrations of brother nodes.

Broken lines in Fig. 4 outline the recursive process of inter-level sorting as an example. Integrating simulative consistent vectors in Fig. 4, we note relative importance of co-nodes in second top level as $W_0 = [0.75, 0.25]$, and continue for other co-nodes as $W_1 = [0.17, 0.83]$, $W_2 = [0.85, 0.15]$, $W_{11} = [0.88, 0.12]$, $W_{12} = [0.20, 0.80]$. Following integrative function gives details of recursive inter-level sorting process.

$$\begin{aligned}
 W_0^T &= ((W_{11}^T \cdot [1, 0] + W_{12}^T \cdot [0, 1]) \cdot W_1^T \cdot [1, 0] + W_2^T \cdot [0, 1]) \cdot W_0^T \\
 \rightarrow W_1^T &= \begin{bmatrix} 0.88 & 0.20 \\ 0.12 & 0.80 \end{bmatrix} \cdot W_1^T = \begin{bmatrix} 0.88 & 0.20 \\ 0.12 & 0.80 \end{bmatrix} \cdot \begin{bmatrix} 0.17 \\ 0.83 \end{bmatrix} \approx \begin{bmatrix} 0.32 \\ 0.68 \end{bmatrix} \\
 \rightarrow W_0^T &= \begin{bmatrix} 0.32 & 0.85 \\ 0.68 & 0.15 \end{bmatrix} \cdot W_0^T = \begin{bmatrix} 0.32 & 0.85 \\ 0.68 & 0.15 \end{bmatrix} \cdot \begin{bmatrix} 0.75 \\ 0.25 \end{bmatrix} \approx \begin{bmatrix} 0.45 \\ 0.55 \end{bmatrix}
 \end{aligned}$$

The final absolute importance of candidate resources to the task in vector $W_0 = [0.45, 0.55]$ can also be expressed as $r_2 > r_1$, which means that the resource r_2 is more suitable than r_1 to run the goal task for evaluating constraints in all levels. It is obvious that most of computing time is occupied by consistency tests in recursions. In other words, the numbers of constraints and candidates are also flexible and scalable in

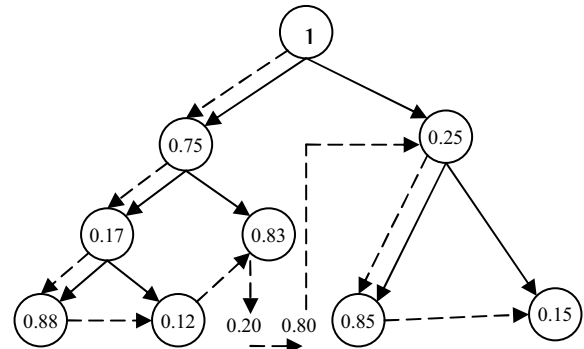


Figure 4. Recursive Inter-level Sorting

MRMT algorithm.

To summarize, we formulate QoS constraints as a MADM problem, and use improved AHP to evaluate absolute importance of candidates. Improved AHP in MRMT algorithm can be used for candidates of both resources and tasks.

C. Priority Matrix with Two-Tuples

As described in section III.B, we can evaluate absolute importance of resources to one task for all QoS constraints, and vice versa. In our work, we build a two-tuples matrix using importance of both resources and tasks. The built two-tuples matrix is called as priority matrix.

The reason of composing a priority matrix with two-tuples is that there may be matching clash happened for multiple tasks. While electing resources for several tasks

synchronously, some tasks may want the same optimal resource. We have to evaluate constraints of resources and make a choice in conflicting tasks now according to preference of resources.

Before building a priority matrix of concurrent tasks and their candidates, a pre-work is sequencing absolute importance vectors of candidates with non-negative integers. As an example, the final vector $W_0 = [0.45, 0.55]$ in section III.B can be sequenced into a sequence as $[2, 1]$. It is visible that we mark the most important candidate as natural number 1, the second most importance as 2, etc. The process of sequencing is similar to sorting methods such as bubble sorting, selection sorting, inserting sorting, etc. Because candidates of multiple tasks are combined together in the priority matrix, and so do tasks, 0 values may appear in result matrix for non-importance or refusal.

There are four concurrent tasks and their candidate resources with corresponding importance as $T_1 = [r_1:0.31, r_3:0.55, r_4:0.14]$, $T_2 = [r_1:0.40, r_2:0.30, r_3:0.20, r_4:0.10]$, $T_3 = [r_1:0.35, r_2:0.38, r_3:0.22, r_4:0.05]$, $T_4 = [r_1:0.37, r_2:0.23, r_3:0.19, r_4:0.21]$. All resources have importance vectors for concurrent tasks as $R_1 = [t_1:0.50, t_2:0.25, t_3:0.15, t_4:0.10]$, $R_2 = [t_1:0.31, t_2:0.45, t_3:0.24]$, $R_3 = [t_1:0.25, t_2:0.23, t_3:0.07, t_4:0.45]$, $R_4 = [t_1:0.47, t_2:0.13, t_3:0.19, t_4:0.21]$. Sequencing all importance vectors of concurrent tasks and resources, we can build the priority matrix in formulation of tasks $\times$ resources, shown as Fig. 5. Horizontals in matrix are concurrent tasks, while verticals are a combination set of candidate resources. There are two values for all elements in the priority matrix: the first values indicate importance of the resources in verticals, and the second values indicate importance of the tasks in horizontals.

Because it is convenient of modelling concurrent tasks, candidate resources and corresponding importance vectors as a weighted bipartite graph, we ever wanted to confirm final matches between tasks and resources by weighted Hungarian algorithm. Hungarian algorithm was given up finally, however, for a reason of wasting computing time and ignoring preference of resources. In order to match between tasks and resources efficiently by Gale-Shapley, a priority matrix must be built firstly in MRMT.

D. Optimization of Matching for Multi-tasks

A priority matrix of concurrent tasks and candidates is built in section III.C, shown as Fig. 5. There are elements of $E_{t_1 r_2}$ and $E_{t_4 r_2}$ who include 0 values for matching relations of t_1 - r_2 and r_2 - t_4 separately. The 0 value at $E_{t_1 r_2}$ indicates that resource r_2 is not a candidate of task t_1 at all, but is candidates of others, and the 0 value at $E_{t_4 r_2}$ is analogous.

There is a clash between tasks t_2 and t_4 for the same candidate resource r_1 , shown as Fig. 5. We decide which task matches with resource r_1 according to the second values (importance of tasks t_2 and t_4 to the resource r_1) in the priority matrix. Of course there also may be a clash between resources r_1 and r_4 for the same task t_1 in theory in Fig. 5. We summarize Gale-Shapley steps in our work from resources with Fig. 5 as follow details.

- 1) r_1 wants t_1 , r_2 wants t_2 , r_3 wants t_4 , r_4 wants t_1 , and t_1 refuses r_4 .
- 2) r_4 wants t_4 , and t_4 refuses r_3 .
- 3) r_3 wants t_1 , and t_1 refuses r_1 .
- 4) r_1 wants t_2 , and t_2 refuses r_2 .
- 5) r_2 wants t_1 , and t_1 refuses r_2 .
- 6) r_2 wants t_3 , and there is no refusal happened, over.

It is visible that the clashes between tasks in Fig. 5 do not appear in our Gale-Shapley steps at all. If we began Gale-Shapley from tasks, however, only clashes between tasks would happen. To abstract this phenomenon in one sentence, if we marked candidates whom the Gale-Shapley began from as active operators, and marked others as passive operators, there were only clashes between active operators in a priority matrix would appear in our Gale-Shapley steps.

The final matched results for concurrent tasks and resources in Fig. 5 are (t_1, r_3) , (t_2, r_1) , (t_3, r_2) , (t_4, r_4) . There are at least two issues presented by the results as follow.

1) As many tasks match optimal resources synchronously, while a few tasks were sacrificed.

2) Under the promise of matching optimal resources for tasks, all matched resources also match tasks as preferable as possible.

There is a feature in our Gale-Shapley steps that active

	r_1	r_2	r_3	r_4
t_1	2,1	0,2	1,2	3,1
t_2	1,2	2,1	3,3	4,4
t_3	2,3	1,3	3,4	4,3
t_4	1,4	2,0	4,1	3,2

Figure 5. Priority Matrix

operators always match less optimal candidates, and passive operators always match more optimal candidates for every refusal happening. Reviewing Gale-Shapley steps from 1 to 6, the more refusals happen, the more optimal candidates matched for passive operators, and the less optimal candidates matched for active operators. It is of course available that we can operate tasks as active operators, and match optimal tasks for resources with our Gale-Shapley steps.

While the process of sequencing absolute importance of candidates with 0 and natural numbers in our work is similar with sorting methods, the computing time of building a priority matrix is always less than $O(n^2)$, where n is the maximum of tasks and resources. Applying Gale-Shapley in the priority matrix, there is at most $n(n+1)/2$ refusals happening for less than $n+1$ tasks and resources. It is said that the algorithm of MRMT in our work is flexible in tasks and resources while we match resources for multi-tasks synchronously.

To sum up, while we match resources for tasks synchronously with MRMT algorithm, QoS constraints are evaluated more objectively by improved AHP method, and optimal candidates are matched synchronously for scalable tasks or resources by Gale-Shapley. Analysing methods and principles in feasibility, the algorithm of MRMT is flexible in

QoS constraints and current tasks. Not only in quantity, but also in types MRMT is very effective and objective to evaluate QoS constraints. For concurrency of tasks, we firstly match optimal resources for as many tasks as possible, and secondly we match resources with tasks as well as possible.

IV. SIMULATION

It is important to compare performance of MRMT with other classical algorithms. Three algorithms of FCFS, RR and MRMT were simulated in our experiments.

All simulations were implemented in grid environment by inheriting and expanding the simulation tool Gridsim. In order to be more generic and precise, 10, 50, 250, 1250 tasks with stochastic size and QoS constraints were created independently. A comparison diagram was drawn, as Fig. 6, to analyse statistics of performance.

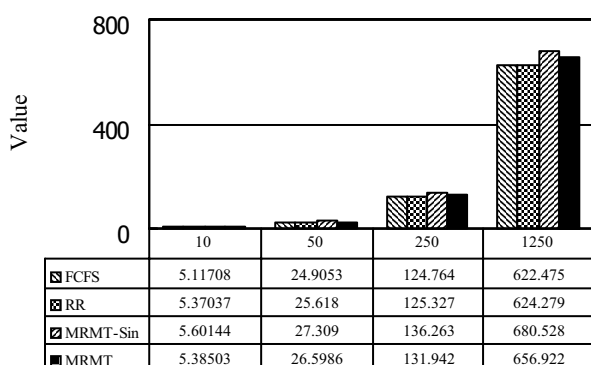


Figure 6. Simulation and Analysis

While all types of QoS constraints are evaluated, optimality of all matched resources is represented by values of y axis, shown as Fig. 6. The larger the value is, the more optimal the matched resources are for all tasks and their all constraints. Comparing bars in Fig. 6, it is obvious that the algorithm MRMT is more optimal than both FCFS and RR. As a particular case of MRMT-Sin, all tasks can match optimal resources, while we match with MRMT linearly.

V. CONCLUSIONS

To improve performance of centreless network, task-resource matching is just one of key items. With improved AHP method, types of scalable QoS constraints are evaluated more objectively while optimal resources are selected out for tasks. With the Gale-Shapley principle and methods, optimal resources are matched synchronously for as many tasks as possible, and at the same time matched resources also are paired with tasks as preferable as possible. As important features, optimal resources always can be selected out for multiple tasks synchronously with MRMT, and matched resources always is optimal in all types of QoS constraints.

There are still some issues for futher study in the algorithm MRMT, such as refreshing values of QoS constraints, modelling dependency of tasks and resources, etc.

ACKNOWLEDGMENT

This work is supported by the China National Science Foundation under Grant 60863015, the "Spring sunshine" Plan of Ministry of Science and Technology of China with project No. Z2007-1-01042, and the Local Science Foundation of Inner Mongolia, China with grant No. 20080404ZD20.

REFERENCES

- [1] Foster I, Kesselman C. The Grid: Blueprint for a New Computing Infrastructure [M]. San Fransisco: Morgan Kaifmann Publishers, 1998.
- [2] Michael Armbrust, Armando Fox, Rean Griffith, etc. Above the Clouds: A Berkeley View of Cloud Computing[R]. Technical Report No. UCB/EECS-2009-28, 2009.
- [3] Flower J. The Future of the Internet: an overview [A]. Cozic, C. P. (ed.). The Future of the Internet [C].San Diego, CA: Greenhaven press, 1997: 10-17.
- [4] Saaty T L. How to Make a Decision: The Analytic Hierarchy Process [J]. European Journal of Operational Research, 1990, 48(1): 9-26.
- [5] Saaty T L. Decision Making: The Analytic Hierarchy and Network Processes (AHP/ANP) [J]. Journal of Systems Science and Systems Engineering, 2004, 13(1): 1-35.
- [6] Braun T D, Siegel HJ, Beck N. A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous processor platforms [J]. IEEE Trans. on Parallel and Distributed Systems, 2004, 15(4): 319-330.
- [7] Vincenzo D M, Marco M. Sub optimal scheduling in a grid suing genetic algorithms [J]. Parallel Computing, 2004, 61(6): 810-837.
- [8] Lin J L, Wu H Z. Scheduling in Grid Computing Environment Based on Genetic Algorithm [J]. Journal of Computer Research and Development, 2004, 41(12): 2195-2199.
- [9] He X S, Sun X H, Laszewski G. QoS Guided Min-min Heuristic for Grid Task Scheduling [J]. Journal of Computer Science and Technology, 2003, 18(4): 442-451.
- [10] Li J, Lu X L, Dong S. Research on Resource Scheduling Strategies for Grid Based on Gridsim[J]. Journal of Computer Science, 2008, 35(8): 95-97.
- [11] Chen J, Pan X. Grid Resource Scheduling Strategy Based on Double QoS Constraints [J]. Computer Integrated Manufacturing Systems, 2008, 14(8): 1571-1578.
- [12] A. K. M. K. A. Talukder, M. Kirley, R. Buyya. Multi-objective differential evolution for scheduling workflow applications on global Grids [J]. Concurrency and Computation: Practice and Experience, 2009. www.interscience.wiley.com
- [13] Xu Y Q, Wang Z J, Xu F. Scheduling Algorithm based on modified QoS Guided Min-Min in Grid [J]. Computer Applications, 2007, 27: 215-216.
- [14] Liu K P, Zhang W Z, Hu M Z. Grid Task Scheduling Algorithm Based on Multiple QoS Constraints [J]. Computer Engineering and Applications, 2006, 29: 107-110.
- [15] Yan T, Li W, Li C Y. QoS Guided Task Scheduling Heuristic in Computational Grid Environments [J]. Microelectronics and Computer, 2006, 23(10): 107-110.
- [16] Lv H, Zhong C, Li Z. Extended QoS-Guided Grid Task Scheduling Algorithm [J]. Computer Engineering, 2007,33(24): 86-88.
- [17] Yang C X, Zhou J C. A Grid Task Scheduling Algorithm Based on Multi-QoS Constraints [J]. Computer Systems and Applications, 2009, (1): 39-43.
- [18] Gong H C, Yu J, Hou Y, Liu H W. User QoS and System Index Guided Task Scheduling in Computing Grid [J]. Computer Engineering, 2009, 35(7): 52-54.
- [19] Wang Y M. An Overview on Ranking Methods of Judgment Matrix [J]. Journal of Decision Making and Decision Support Systems, 1995, 5(3): 101-114.