

Journal of Zhejiang University SCIENCE A
 ISSN 1009-3095 (Print); ISSN 1862-1775 (Online)
 www.zju.edu.cn/jzus; www.springerlink.com
 E-mail: jzus@zju.edu.cn



Resources publication and discovery in manufacturing grid*

TAO Fei[†], HU Ye-fa, DING Yu-feng, SHENG Bu-yun, ZHOU Zu-de

(School of Mechanical and Electronic Engineering, Wuhan University of Technology, Wuhan 430070, China)

[†]E-mail: taofei@mail.whut.edu.cn

Received Dec. 18, 2005; revision accepted Feb. 14, 2006

Abstract: In the manufacturing grid's architecture, Resources Management System (RMS) is the central component responsible for disseminating resource information across the grid, accepting requests for resources, discovering and scheduling the suitable resources that match the requests for the global grid resource, and executing the requests on scheduled resources. In order to resolve the problem of resources publication and discovery in Manufacturing Grid (MGrid), the classification of manufacturing resources is first researched after which the resources encapsulation class modes are put forward. Then, a scalable two-level resource management architecture is constructed on the model, which includes root nodes, domain nodes and leaf nodes. And then an RMS is proposed, and the resources publication and discovery mechanism are detailedly described. At last, an application prototype is developed to show the validity and the practicability of the proved theory and method.

Key words: Manufacturing Grid (MGrid), Resources Management System (RMS), Resource classification, Resource publication, Resource discovery

doi:10.1631/jzus.2006.A1676

Document code: A

CLC number: TP18; TH165

INTRODUCTION

Manufacturing Grid (MGrid) is a new concept proposed to meet the practical demands in the manufacturing industry and is a new technology enabling broad geographical distribution of all sorts of manufacturing resources through the Internet or other specific nets, using grid technology. With the MGrid platform, we can realize common sharing of manufacturing resources including human resources, equipment resources, material resources, applicable software systems, etc., and can also realize the cooperative design and manufacture of the same product in different places and enhance the competitive capacity of enterprises by shortening the exploring and manufacturing period of a product, and minimizing the entire cost as well (Shi *et al.*, 2004).

In the grid's architecture, grid Resource Management System (RMS) is the central component

responsible for disseminating resources information across the grid, accepting requests for resources, discovering and scheduling suitable resources that match the requests from the global grid resources, and executing the requests on scheduled resources (Chen *et al.*, 2004; Krauter *et al.*, 2002).

Many projects have addressed the problem. Systems such as I-SOFT (Vahdat, 1999), PBS (Berman and Wolski, 1997), LSF (Kyunqkoo *et al.*, 2000), NQE (Ekmecic *et al.*, 1996), and Load Leveler (IBM Corporation, 1993) process the user submitted jobs by finding resources identified either explicitly through a job control language or implicitly by submitting the job to a particular queue that is associated with a set of resources, this manually configured queue hinders the dynamic resource discovery.

Globus (Tuecke *et al.*, 2003) provides grid information services via an LDAP-based network directory called Metacomputing Directory Services (MDS) consisting of two components: Grid Index Information Service (GIIS) and Grid Resource Information Service (GRIS). Globus presents resource management architectures that support resource dis-

* Project supported by the National Natural Science Foundation of China (No. 50335020) and the Opening Foundation of Hubei Digital Manufacturing Key Lab (No. SZ0406), China

covery, dynamical resource status monitor, resource allocation, and job control.

The AppLeS project (Berman and Wolski, 1997) primarily focuses on developing scheduling agents for individual applications on production computational grids and uses the services of other resources managements such as Globus, Legion. AppLeS agents use application and system information to select a viable set of resources. And the AppLeS framework (Liu *et al.*, 2002) guides the implementation of application-specific scheduler logic, which determines and actuates a schedule customized for the individual application and the target computational grid at execution time. However, AppLeS schedulers do not offer QoS support.

Condor (Litzkow *et al.*, 1998; Raman *et al.*, 1998; Frey *et al.*, 2001) is a high-throughput computing environment that can manage a large collection of diversely owned machines and networks, and is well known for harnessing idle computers. It provides a general resource selection mechanism based on ClassAd language, using a semi-structured data model for resource description and allowing users to describe arbitrary resource requests and resource owners to describe their resources. A matchmaker is used to match user requests with appropriate resources in Condor. Because the ClassAd language and the matchmaker are designed for selecting a single machine on which to run a job, so they cannot easily be applied when a job requires multiple resources.

Most of the systems described above are mainly about the computational resources, such as computer, PC, storage resources, etc. The MGrid, however, very different from those computational grid systems, pays attention to computational resources, but even more attention to manufacturing resources. It is mainly designed to solve the sharing and cooperating of manufacturing resources, such as equipments resources (machine tool, clamp, etc.), application resources (CAPP/CAD/CAM, ERP, PDM, ANALYSE, etc.), human resources (manager, worker, technician, designer, salesman, etc.), material resources (semi-manufactured goods, raw and processed materials, fuel, etc.). So resources publication and discovery are more complicated and difficult in MGrid system than in computational grid system.

In order to realize the common sharing of various resources in MGrid, and optimally reorganize the

resources, which eventually lower costs and shorten manufacturing time, the classification of manufacturing resources is researched, and the resources encapsulation class modes are put forward. Then, a scalable two-level resource management architecture is constructed upon the model, which includes root nodes, domain nodes and leaf nodes. The resources management system is proposed, and the resources publication and discovery mechanism in manufacturing is detailedly described at last.

CLASSIFICATION AND ENCAPSULATION MODEL OF RESOURCES IN MGRID

In manufacturing grid, resources differ greatly from each other in terms of the nature (e.g. physical characteristics, location, dynamicity, sensitivity, functionality), the demands placed on (e.g. time, quality, cost, service), and the ways in which they are engaged (e.g. discovery, brokering, monitoring, diagnosis, adaptation) (Iamntichi and Foster, 2001). But, we can take resources of similar characteristics and demands as the same category. So, in order to enhance the efficiency of management and service of MGrid resource, and to improve the flexibility, expansion and fault-tolerance of MGrid system, we classify the resources into 9 categories according to the nature, the demands placed on, and the ways in which they are engaged. The 9 categories are: (1) Human Resources (HR), (2) Equipment Resources (ER), (3) Material Resources (MR), (4) Application System Resources (ASR), (5) Technology Resources (TR), (6) Service Resources (SR), (7) Computer Resources (CR), (8) User Information Resources (UIR), and (9) Other Resources (OR). Then we encapsulate each kind of resources into a resources class model according to related resources categories, the nine encapsulation class models of manufacturing resources in MGrid are given in Table 1.

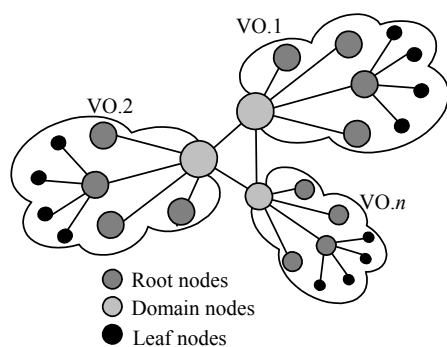
RESOURCE MANAGEMENT ARCHITECTURE BASED ON MANUFACTURING RESOURCE CLASSIFICATION

According to the manufacturing resource classification model and corresponding resources encapsulation

Table 1 The key encapsulation class of manufacturing resources in MGrid

Name	Definition	Instance
HR-class	The people that have certain ability required in manufacturing activities, such as operation, manage or technique and so on	Designer, manager, technician, etc.
ER-class	All kinds of equipments that have certain function required in manufacturing activities	Machine tool, CNC, clamp, measure, etc.
MR-class	Material, semi-manufactured goods, raw and processed materials, fuel, finished-product and so on required in manufacturing activities	Material, raw and processed materials, fuel, product, etc.
ASR-class	All kinds of software required in manufacturing activities, including design systems, simulation systems, analyzing systems, management systems, graph processing systems, etc.	CAD, CAPP, CAE, PDM, ERP, ANALYSE, Pro/E, Photoshop, Coreldraw, Solidworks, etc.
TR-class	The set of technical resources and condition required in manufacturing activities	Drawing, design flow, process planning flow, etc.
SR-class	Information querying, training, maintenance and so on, provided for resources users	International standards, national standards, etc.
CR-class	Computational resources required in manufacturing activities	CPU, storage, etc.
UIR-class	The information that records the basic information about both resources provider and user, such as credit, etc.	Enterprise information, user information, etc.
OR-class	Other resources which are not included in the above classes, such as log, account information and public information	Log, account, etc.

sulation class model in MGrid shown before, we classify the resources into nine self-government domain in a Virtual Organization (VO) of MGrid. We have named each domain as domain node (e.g., HRD: Human Resources Domain; ERD: Equipment Resources Domain; MRD: Material Resources Domain, and ASRD: Application System Resources Domain, etc.) in a VO, the node that manages all the domain nodes in a VO as root node, the node which stands for resources such as machine, computer, clusters under other domain nodes as leaf node. So, a scalable two-level resource management architecture of manufacturing resource in MGrid is formed, as shown in Fig.1.

**Fig.1 Resource management architecture in MGrid**

Root node mainly consists of information on the other VO's root node and information on domain nodes. The former is mainly responsible for describing the root node information of other virtual organizations and responds to service requests, allocating them to their domain nodes. And the later (information on domain nodes) governs the basic information on every domain node. Then domain node manages the resource information published in it.

RESOURCES MANAGEMENT SYSTEM IN MGRID

The enterprises and resources in MGrid are heterogeneous, dynamic, autonomous and geographically distributed. So the RMS in MGrid should satisfy the following main requirements: (1) encapsulating all sorts of manufacturing resources; (2) networked resources publishing; (3) resources discovering based on manufacturing task; (4) resources optimal selection evaluating; (5) resources services trust evaluating. According to these requirements, we have designed an RMS in MGrid, which is shown in Fig.2. The proposed RMS can provide fundamental functions for remote resources encapsulation, registry, discovery, optimal resources selection, trust evaluating and monitoring by integrating MDS and the other modules provided by Globus.

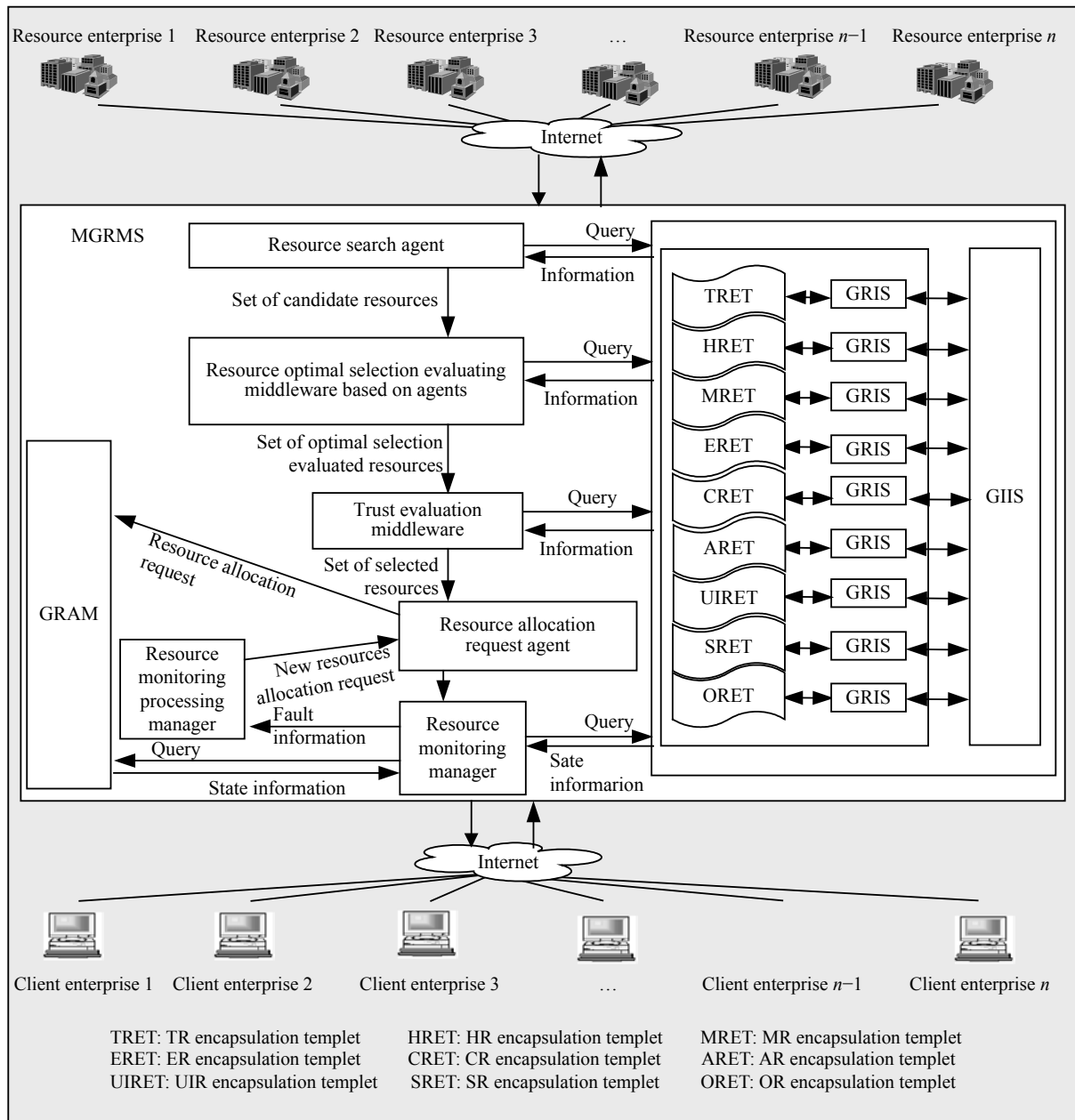


Fig.2 Resource management system in MGrid

The main components of the proposed RMS include a resources search agent, a resources optimal selection evaluating middleware based on agents, a trust evaluation middleware, a resource allocation request agent, resources monitoring manager, resources monitoring processing manager.

(1) Resources search agent: resources search agent is responsible for discovering the resources meeting the requirements described by a client (user) from the resources encapsulation templates and gene-

rates the candidate set of resources.

(2) Resources optimal selection evaluating middleware based on agents: provides comprehensive evaluation to the candidate resources and selects the optimal set of resources suitable for the manufacturing task, which can improve the efficiency and quality of the resource selecting and the service quality as well. The specific evaluation mode and algorithms were researched in (Tao et al., 2005).

(3) Trust evaluation middleware: gets the trust

relationship among users and resources providers in MGrid, and provides mechanisms for trust evaluation and trust updating for desirable trust dynamics to allow trust decision making. Research of specific trust evaluation mode and algorithms are described in "Modelling and application on trust evaluation for manufacturing grid resources service".

(4) Resources allocation request agent: transforms the manufacturing task into a special language for resources allocation and requests GRAM to allocate resources.

(5) Resources monitoring manager: monitors the resource states, decides the occurrence of a failure by analyzing state information of each resource and communicates with the resources monitoring processing manager.

(6) Resources monitoring processing manager: displays the state of resource and the type of failure that occurred to the user, and requests whether the job executed by the resources where failure occurred should be migrated and rescheduled or not.

RESOURCE PUBLICATIONS AND DISCOVERY IN MGRID

Grid resource publication process proceeds as follows:

(1) The resource provider (the resource enterprise) submits the requests of resources or service publication to a VO's root node in MGrid via the human-machine interface through Internet if intending to join the VO.

(2) MGrid verifies the identity and validity of the provider and delivers the request and result to the corresponding domain node responsible for registration.

(3) The corresponding domain then returns the successful message to the root node, encapsulating the resources into the corresponding resource encapsulation template if the registration is valid, and returns unsuccessful message if resource registration failed in the other case.

(4) The root node returns the GSH to the corresponding resource provider.

(5) The resource provider will periodically send out the living messages to the corresponding domain, called soft-state registration. At the same time, the

specific domain may subscribe with the resource provider for some metadata. When the subscribed metadata changed, the resource provider will notify the domain to register it again in time.

The resource discovery is implemented as follows:

(1) The user (client enterprise) submits a manufacturing task or resources service requirements to MGrid via the human-machine interface through the Internet.

(2) The resource search agent uses the resource encapsulation model, GRIS and GISS of Globus which provide the information about resources, generates the set of candidate resources, and submits it to resource optimal selection evaluating middleware based on agents.

(3) The resource optimal selection evaluating middleware based on agents (Tao *et al.*, 2005) evaluates the candidate resources by running the algorithm employed in the evaluation model to select the set of optimal resources for efficient and economical job execution, and generates a set of optimal selection evaluated resources that are then submitted to trust evaluation middleware.

(4) The resources trust evaluation middleware evaluates the trust relationship between the user and the candidate resources. From the user's viewpoint, it should select the best trustworthy resources the user holds to faithfully and successfully complete the job request. From the resources provider's viewpoint, it should provide the evidence and reference to decide whether the user can validly use the resources to complete the task, such as producing good manufacturing task code, etc. And the trust evaluation middleware returns the set of selected resources to resource allocation request agent after evaluation.

(5) The resource allocation request agent requests GRAM to allocate the resources. And the GRAM informs the resources providers accordingly.

(6) At the same time, the resources monitoring manager monitors the resources states, and informs the resources monitoring processing manager if some failures occur in the resources.

(7) The resources monitoring processing manager displays the state of resources and the type of failure that occurred to the user and requests the GRAM to reallocate resources for the job.

(8) Repeat Steps 5~7.

The detailed sequence diagrams of resource publication and discovery are omitted due to the limited paper space.

APPLICATION PROTOTYPE

In practice we take the processing of spindle sleeve in MGrid as an example to test the validity and practicability of the theory and the method proposed above, and have developed an application prototype. The brief flow of spindle sleeve processing in MGrid is stated below: (1) Resource enterprise publishes its resources via RMS in MGrid, and Fig.3 is the publishing interface of resources of machine tools which belong to ER in MGrid; (2) The client enterprises define and describe manufacturing job via the job management system interface in MGrid, as shown in Fig.4. And the job model will be generated after the validity of the defined job is checked; (3) The resource search agent uses the resource encapsulation model, GRIS and GIIS of Globus, which provides the information about resources and generates the set of candidate resources according to the job mode, and then submits it to resource optimal selection evaluating middleware based on agents. The optimal selection evaluating results of spindle sleeve processing resources are shown in Fig.5; (4) Repeat Steps 4~7 described in the resource discovery process in the previous section.

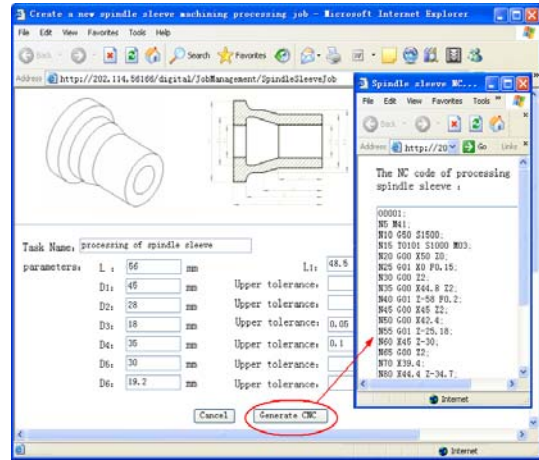


Fig.4 Processing job definition of the spindle sleeve

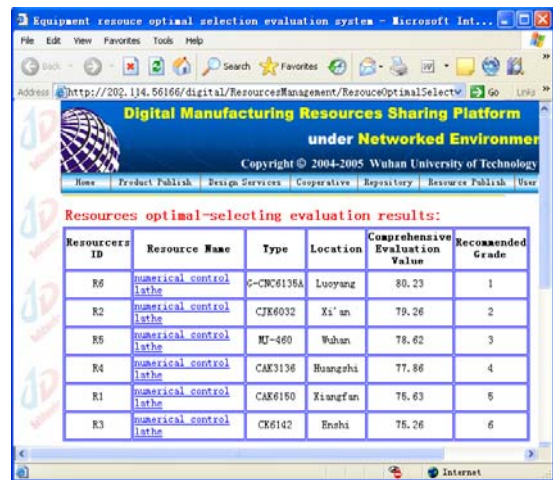


Fig.5 Optimal selection evaluating results of the spindle sleeve processing resources

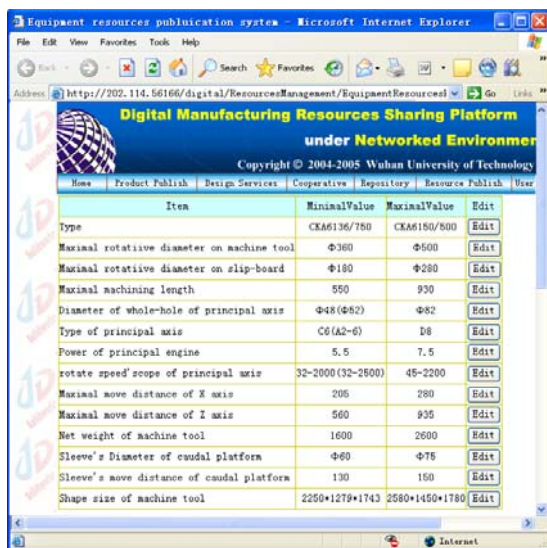


Fig.3 Publication and encapsulation interface of the machine tools

CONCLUSION AND FUTURE WORK

Through research of classification of MGrid manufacturing resources, presenting the classification model, we propose a scalable two-level MGrid resource management architecture, and also give a thorough description of the resource publication and discovery mechanism in this resource management architecture. There are still more to be researched in MGrid, and our study is only a beginning. In the future, we will work on transaction and safety problems of the resource-service in the MGrid, and the metadata and metaservice of MGrid resource.

References

- Berman, F., Wolski, R., 1997. The AppLeS Project: A Status Report. Proceedings of the 8th NEC Research Symposium. Berlin, Germany.
- Chen, X.L., Yang, C., Lu, S.L., Chen, G.H., 2004. An Active Resource Management System for Computational Grid. The 3rd International Conference on Grid and Cooperative Computing (GCC'2004). Wuhan, **3251**:225-232.
- Ekmecic, I., Tartalja, I., Milutinovic, V., 1996. A survey of heterogeneous computing: concepts and systems. *Proceedings of the IEEE*, **84**(8):1127-1144. [doi:10.1109/5.533958]
- Frey, J., Tannenbaum, T., Livny, M., Foster, I., Tuecke, S., 2001. Condor-g: A Computation Management Agent for Multi-institutional Grids. Proceedings of the 10th IEEE International Symposium on High Performance Distributed Computing. San Francisco, California, p.55-63.
- Iamntichi, A., Foster, I., 2001. On Fully Decentralized Resource Discovery in Grid Environments. Proceedings of the 2nd International Workshop on Grid Computing. London, UK, p.51-62.
- IBM Corporation, 1993. IBM Load Leveler: User's Guide. Kingston, NY, USA. <http://lscftp.kng.ibm.com/ppps/products/loadleveler.html>
- Krauter, K., Buyya, R., Maheswaran, M., 2002. A taxonomy and survey of grid resource management systems. *International Journal of Software, Practice and Experience*, **32**(2):135-164. [doi:10.1002/spe.432]
- Kyunqkoo, J., Ladislau, B., Krzysztor, P., Marinescu, D.C., 2000. Agent-based Resource Discovery. Proceedings 9th IEEE Heterogeneous Computing Workshop, p.43-52.
- Litzkow, M., Livny, M., Mutka, M., 1998. Condor—A Hunter of Idle Workstations. Proceedings of the 8th International Conference of Distributed Computing Systems, p.104-111.
- Liu, C., Yang, L.Y., Foster, I., Angulo, D., 2002. Design and Evaluation of a Resource Selection Framework for Grid Applications. Proceedings of the 11th IEEE International Symposium on High Performance Distributed Computing HPDC-11 2002 (HPDC'02). Washington DC, USA, p.63-72.
- Raman, R., Livny, M., Solomon, M., 1998. Matchmaking: Distributed Resource Management for High Throughput Computing. IEEE International Symposium on High Performance Distributed Computing. San Francisco, California, USA, p.140-146.
- Shi, Z.B., Yu, T., Liu, L.L., 2004. Manufacturing grid and its resource configuration algorithm. *Computer Engineering*, **30**(5):117-119 (in Chinese).
- Tao, F., Hu, Y.F., Ding, Y.F., Sheng, B.Y., Zhou, Z.D., 2005. Resources optimization selection evaluation model based on agents in manufacturing grid system. *China Mechanical Engineering*, **16**(24):2192-2197 (in Chinese).
- Tuecke, S., Czajkowski, K., Foster, I., Frey, J., Graham, S., Kesselman, C., Maguire, T., Sandholm, T., Vanderbilt, P., Snelling, D., 2003. Open Grid Services Infrastructure (OGSI), Version 1.0. Global Grid Forum Draft Recommendation.
- Vahdat, A., 1999. Toward Wide-area Resource Allocation. Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications, 2:930-936.

Welcome visiting our journal website: <http://www.zju.edu.cn/jzus>

Welcome contributions & subscription from all over the world

The editor would welcome your view or comments on any item in the journal, or related matters

Please write to: Helen Zhang, Managing Editor of JZUS

E-mail: jzus@zju.edu.cn Tel/Fax: 86-571-87952276/87952331