Proceedings of the 8th WSEAS Int. Conference on Automatic Control, Modeling and Simulation, Prague, Czech Republic, March 12-14, 2006 (pp213-216)

A Grid Dynamic Analyzing Resource Model

HUEY-MING LEE, CHENG-HSIUNG YANG, TSANG-YEAN LEE, MU-HSIU HSU, YAO-JEN SHIH, PIN-JEN CHEN
Department of Information Management
Chinese Culture University
55, Hwa-Kung Road, Yang-Ming-San
Taipei (11114), TAIWAN

Abstract: - Grid computing architecture was defined to be a complete physical layer. The functions of information systems based on grid architecture are resources sharing, collaborative processing, etc. However, each resource of collaborated grid nodes changes dynamically, how to make these resources optimize plan and allocation is an important issue. In this paper, we proposed a dynamic analyzing resources model based on grid computing architecture. This model can make the plans and allocations of the resources of collaborated nodes optimize. Via the experiment of this model, we can have that the resource allocations are more efficient.

Key-Words: - Grid computing; Resources allocation

1 Introduction

The term "Grid" was coined in the mid 1990s to denote a proposed distributed computing infrastructure for advanced science and engineering [8]. In grid environment, users may access the computational resources at many sites [5, 9]. The functions of information systems based on grid computing architectures are resources (e.g., CPUs, storages, etc.) sharing, collaborative processing, reliable and secure connection, etc. However, each resource of coordination node in the grid environment, e.g., CPU loading, memory rate of utilization, etc. changes dynamically, how to optimize these resources allocation is an important issue.

Many researchers have addressed the resource allocation problem. Condor [11] provides a general resource selection mechanism base on the ClassAds language [14], which allows users to describe arbitrary resource requests and resource owner to describe their resources. Raman et al. [13] developed and implemented the classified advertisement (classad) matchmaking framework which was designed to solve real problems encounter in the deployment of Condor. The matchmaker is used to match user requests with appropriate resources.

Because the classAds language and the matchmaker were designed for selecting a single machine on which to run a job, however, it has limited applicability in the situation where a job requires multiple resources [12]. To address these problems, Liu et al. [12] defined a set-extended ClassAds Language that allows users to specify aggregate resource properties (e.g., total memory, minimum bandwidth. They also presented an extended set matching matchmaking algorithm that supports one-to-many matching of set-extended ClassAds with resources. Based on this technique, they presented a general-purpose resource selection framework that can be used by different kinds of applications. Foster et al. [7] presented the grid resource allocation and management (GRAM). GRAM simplifies the use of remote systems by providing a single standard interface for requesting and using remote system resources for the execution of "jobs". The most common use of GRAM is remote job submission and control. GRAM is designed to provide a single common protocol and API for re-questing and using remote system resources, by provide a uniform, flexible interface to local job scheduling systems.

Moreover, some authors presented the computation economy for the grid to solve the resource allocation problem. Chun [2] proposed a market-based approach to cluster management based on the notion of a computational economy which optimizes for user value. Wolski et al. [17], investigated G-commerce – computational economies for controlling resource allocation in computational grid settings. Buyya [1] proposed a distributed computational economy as an effective metaphor for the management of resources and application scheduling.

In this study, we proposed a grid dynamic analyzing resources model. It can automatically detect the resources of grid nodes and optimize allocations of the resource according to the different users' requests. Via the experiment of this model, we can have that the resource allocations are more efficient.

2 Framework of the Proposed Model

In this section, we presented a dynamic analyzing resources model (DARM) based on grid computing architecture. It was built on the grid node, saying M0, as shown in Fig. 1.

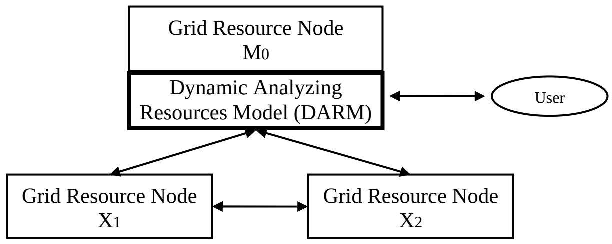

Fig. 1 Context of the proposed model (DARM)

There are four modules in this model, saying, user interaction module (UIM), data analysis module (DAM), computation analysis module (CAM), and job manipulation module (JMM), as shown in Fig. 2. The functions of these modules are as the follows:

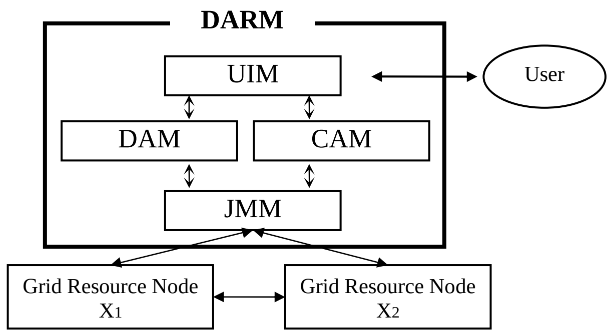

Fig. 2 Framework of the proposed model (DARM)

• User interaction module (UIM): It can receive the user's proceeding requests (e.g., data files, or computation, etc.) and transfer the executed results to data analysis module (DAM) or computation analysis module (CAM) depending on the request type, and to user.

• Data analysis module (DAM): It can divide the data files into some replicas, and store them to different grid nodes.

• Computation analysis module (CAM): There are three kinds of different allocation methods, saying, average allocation, cost optimization allocation and time optimization allocation for user to select the suitable one depending on requests of processing.

• Job manipulation module (JMM): It can explore the resources of the grid notes, and distribute works to others nodes.

2.1 User interaction module (UIM)

After receiving the requests from the user, user interaction module (UIM) can receive the user's processing requirements (e.g. data files, or computation, etc.) and transfer the executed results to data analysis module (DAM), or computation analysis module (CAM) depending on the requirement type, and to user. There are three components in this module, saying, user interaction component (UIC), computation analysis interaction component (CAIC) and data analysis interaction component (DAIC), as shown in Fig. 3.

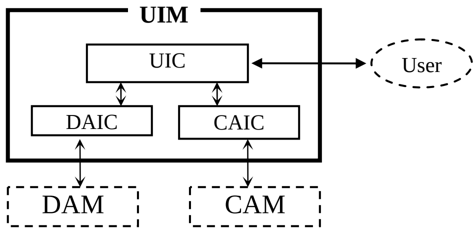

Fig. 3 Framework of UIM

2.2 CAM

There are two components in CAM, saying, resource assessing component (RAC) and resources planning component (RPC), as shown in Fig. 4. Based on [15], we built up a prices listing table for the classifying resources. RAC can receive the re-sources status (e.g., CPU, storages, etc.) via exploring the resources of grid nodes, classify these messages to prices listing table for the attributes. RPC will optimize the resources allocation by linear programming [16], and RPC will pass the optimized results to JMM.

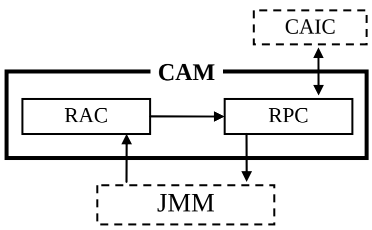

Fig. 4 Framework of CAM

2.3 DAM

DAM comprises of three components, saying, data maintain component (DMC), data record management component (DRMC), data unity component (DUC), and index repository (IR), as shown in Fig. 5.

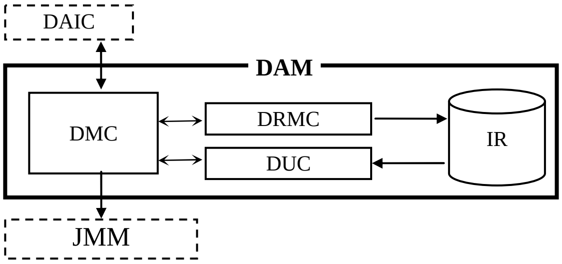

Fig. 5 Framework of DAM

DMC can divide, access, and update the distributed files according to user's requests. DRMC records the messages of dividing replicas into IR. IR stores the messages of the dividing and accessing files, and provides these messages to DUC. DUC accesses the messages of the dividing from IR, and records the processes of updating replicas. When the user wants to access the portion of required files, he/she should access the related replicas via DRMC and DUC. For more detail, please ref. Lee et al. [10].

2.4 JMM
The JMM comprises of two components, saying, resources supervising component (RSC) and authenticate component (AC). AC can authenticate each collaborated grid node and construct security connection. RSC will dynamically supervise the resources of these collaborators.

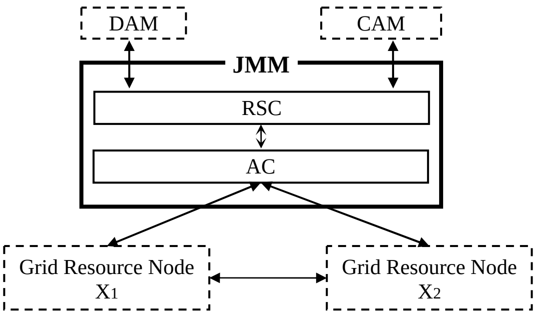

Fig. 6 Framework of JMM

3 Model Implementation
For easy to manipulating, crossing platform and remote controlling, we applied Java Server Page (JSP) and Java Servlet written Web Server structure, Java 2 Platform Standard Edition v 1.4.2 API, Globus toolkit to construct the proposed model. The mentioned above is done by grid computing architecture with three Pentium 4, 2.4GHz computers that are powered by OS Linux Red Hat 9.0, 256MB RAM, a 100-Mbps network interface card, and 100-Mbps Ethernet. Each of computers is connected to HUB which is ZOT IA700. The network environment was run through a LAN that reduced delay and loss of data files. First, we detect the resources of the collaborated grid nodes, the user interface is shown in Fig. 7, and the results after detecting are shown as in Fig. 8.


Fig. 7 The user interface

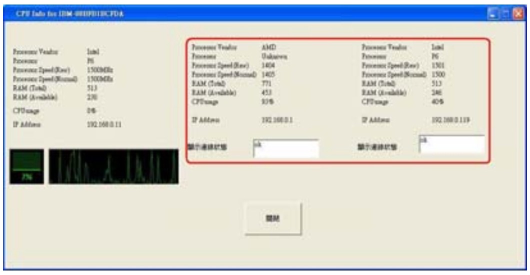
Fig. 8 The results after detecting

After detecting the resource messages of the grid nodes, there are three allocation methods we can choose due our requirement, saying, average allocation, optimizing time allocation and optimizing cost allocation. The DARM will allocate the work to proper nodes according to the user demands. Under the same work type, the computed results by the three allocation methods are shown in Table 1.

Table 1 The computed results by the three allocation methods

Allocation	Executed time (sec)	Executed cost (G-coin)
Average	8.322	1000
Optimizing time	5.968	1100
Optimizing cost	12.728	500

According to Table 1, we allocate work to proper nodes. The cost is lower, the executed time is higher. In other words, if we want to save cost, the executing time will be longer.

4 Conclusion

At present, the requirement of high performance computing is increasing at high speed. Grid computing is getting more and more significantly. However, each re-source of collaborated nodes in the grid computing environment changes dynamically, the dynamic analyzing resources mechanism is one of the most important issues in the grid computing environment.

In this study, we proposed a dynamic analyzing resources model based on grid computing architecture. Via this model, we not only can make the plans and allocations of the resources of collaborated grid nodes optimize, but also can make the resource allocations more effective.

Acknowledgements

This work was supported in part by the National Science Council, Republic of China, under grant NSC94-2745-M-034-008-URD.

References:

[1] R. Buyya, *Economic-based distributed resource management and scheduling for grid computing*, Doctor of Philosophy, School of Computer Science and Software Engineering, Monash University, Melbourne, 2002.

[2] B. Chun, & D. Culler, *Market-based Proportional Resource Sharing for Clusters*, University of California at Berkeley, Computer Science Division, Technical Report CSD-1092, 1999.

[3] J. Ferng, UAGrid [Online]. Available: http://computing.arizona.edu/uagridinternet2day.pdf, 2004.

[4] L. Ferreira, B. Jacob, S. Slevin, S. Sundararajan, M. Brown, J. Lepesant, & J. Bank, Globus toolkit 3.0 quick start, *IBM RedPaper*, 2003, pp. 23-63.

[5] I. Foster, C. Kesselman, Globus: A Metacomputing Infrastructure Toolkit, *International Journal of Supercomputer Application*, Vol.11, No.2, 1997, pp. 115-128.

[6] I. Foster, C. Kesselman, & S. Tuecke, The Globus alliance globus toolkit [Online]. Available: http://www.globus.org/ [1998, May 31].

[7] I. Foster, C. Kesselman, & S. Tuecke, GRAM: Key concept [Online]. Available: http://www-unix.globus.org/toolkit/docs/3.2/gram/key/index.html [1998, July 31].

[8] I. Foster, C. Kesselman, (ed.) *The Grid2: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, San Francisco, 2004.

[9] Globus: http://www.globus.org, 2005.

[10] H-.M. Lee, and C.-H. Yang, *A Distributed Backup Agent Based on Grid Computing Architecture, Knowledge-Based Intelligent Information & Engineering Systems*, LNCS 3682/2005, Springer-Verlag, 2005, pp. 1252-1257.

[11] M. Litzkow, M. Livny, and M. Mutka, Condor-A Hunter of Idle Workstations, *in Proceedings of the 8th International Conference of Distributed Computing Systems*, 1998.

[12] C. Liu, L. Yang, I. Foster, D. Angulo, Design and Evaluation of a resource Selection Framework for Grid Applications, *in Proceedings of IEEE International Symposium on High Performance Distributed Computing*, 2002, pp. 63-72.

[13] R. Raman, M. Livny, & M. Solomon, Matchmaking: Distributed Resource Management for High Throughput Computing. *In Proceedings of the Seventh IEEE International Symposium on High Performance Distributed Computing*, 1998, pp. 140-146, November 4,

[14] R. Raman, ClassAds Programming Tutorial (C++), 2000.

[15] Research, Development, and Evaluation Commission, Executive Yuan: The Information Outsourcing Promotion of the Government Department, December 15, 2003, Available: http://web.rdec.gov.tw/cisa/CaseRule_Profile.htm?mdy=14.

[16] W. L. Winston, *Operations Research*, Duxbury Press, Belmont, California, Third Edition, 1994.

[17] R. Wolski, J. Plank, J. Brevik, & T. Bryan, G-commerce: Market Formulations Controlling Resource Allocation on the Computational Grid. *In Proceedings of the 15th International Parallel & Distributed Processing Symposium*, 2001, pp. 46.