PCGrid: Integration of College's Research Computing Infrastructures Using Grid Technology*

Kuan-Ching Li[1] Chiou-Nan Chen[1, 2] Chun-Chieh Liu[1] Chia-Fu Chang[1] Chia-Wen Hsu[1]
Sheng-Shiang Hung[1] Chun-Yu Lai[1]

[1] PDPC - Parallel and Distributed Processing Center
Dept. of Computer Science and Information Management
Providence University
Shalu, Taichung 43301 Taiwan.
E-mail: {ccliu, cfchang, cwhsu, sshung, kuancli}@cs.pu.edu.tw

[2] Laboratory of Bioinformatics and Computational Biology
Department of Computer Science
National Tsing Hua University
Hsinchu City, Hsinchu 30013 Taiwan
E-mail: cnchen@cs.nthu.edu.tw

Abstract — **Internet computing and grid technologies promise to change the way we tackle complex problems. Grid computing environments are characterized by interconnecting a number of heterogeneous hosts through the Internet. They enable large-scale aggregation and sharing of computational, data and other resources across institutional boundaries. In this paper, it is introduced a grid computing platform named PCGrid, which basically is built by interconnecting a number of high-end cluster computing resources and sharing available resources in College of Computing and Informatics at Providence University, Taiwan. The main goal for constructing such platform is to perform investigations in system technologies and high performance applications, such as medical images, bioinformatics, high-energy physics, and system level APIs. This novel project shows the viability of such implementation inside a campus.**

Keywords— Grid computing, computing infrastructure, educational platform.

I. INTRODUCTION

A computational grid is a collection of distributed and heterogeneous computing nodes that has emerged as an important platform for computation intensive applications [1-7, 24]. It offers a model for solving massive computational problems using large numbers of computers, arranged as clusters embedded in a distributed infrastructure.

Grid computing involves the sharing of heterogeneous resources (based on different platforms, hardware/software, computer architecture, computer languages), located at different places and belonging to different administrative domains over a network. That is, it involves revitalizing computing resources distributed in different locations. The development and deployment of computational Grids is focused on creating a more seamless and direct means of utilizing computationally based resources.

The successful implementation of Grid computing infrastructures will certainly have far-reaching implications for the business, scientific and individual computing users. Different computational science researches, as also for commerce, industrial and social service applications can certainly benefit of such technology. For instance, an organization can pool all its computing resources spread across different locations and create a super computing environment. A researcher who needs high-speed machine with massive storage for some rigorous computing intensive projects does not need to procure costly equipment; he/she can simply hire the grid provider's computing resources.

The remainder of this research paper is organized as follows. Section 2 discusses core technologies for the development of this project. Section 3 introduces PCGRID, a grid platform built up by interconnecting a number of computational resources located in different laboratories inside Providence University (PU)'s College of Computing and Informatics (CCI). Later in section 4, we introduce ongoing research projects being investigated by our research teams using such platform. Finally, in section 5, some conclusions and future works are presented.

II. CORE TECHNOLOGIES

In this section, it is introduced core technologies are fundamentals to build the PCGrid grid platform. In subsection 2.1, grid-related core technologies are introduced, in subsection 2.2 the Globus toolkit, and finally in subsection 2.3 the MPICH-G2 is introduced.

* This research is supported in part by National Science Council, Taiwan, under grants NSC93-2213-E-126-010, NSC94-2213-E-126-005, and National Center for High Performance Computing, Taiwan, for sponsoring the Taiwan Unigrid Project, under "Taiwan Knowledge Innovation National Grid" Project.

A. Grid Technologies

The grid architecture is an extensible and open structure organized in component layers [8-14, 19], within which solutions to key user requirements can be placed. Components within each layer share common characteristics, and it is able to build capabilities and behaviors present in any lower layer. Thus, the architectural description is high level and places few constraints on the design and implementation. The layered grid architecture and its relationship to the Internet protocol architecture is shown in Figure 1.

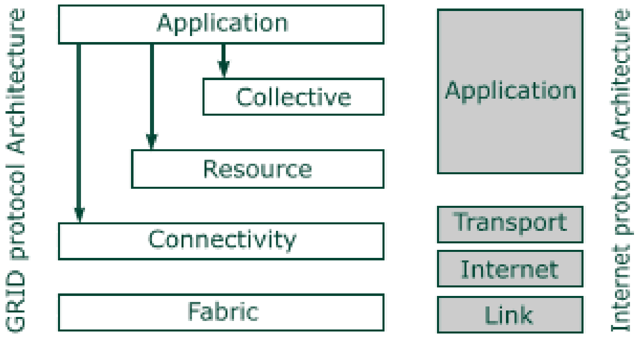


Fig. 1. The grid architecture.

The Grid Fabric layer contains the resources that are to be shared, which include computational power, data storage, sensors, among others. This sharing is controlled by grid protocols, and the resources can include local networks. In this case, the local protocols take over at this point. One layer above, the Connectivity layer contains the communication and authentication protocols required for Grid-specific network transactions. Communication protocols enable the exchange of data between different Fabric layer resources, while authentication protocols build on communication services to provide secure mechanisms for verifying the identity of users and resources.

The Resource layer uses the communication and security protocols of the Connectivity layer to control the secure negotiation, initiation, monitoring, control, accounting, and payment of sharing operations on individual resources [15, 18, 20]. Resource layer protocols call Fabric layer functions to access and control local resources. Resource layer protocols are concerned entirely with individual resources. While the Resource layer is focused on interactions with single resources, the Collective layer contains protocols and services that are global in nature and capture interactions across collections of resources. Collective components are designed that they implement a wide variety of sharing behaviors without placing new requirements on the fabric resources being shared such as: A directory service may allow users to query for resources by name or by attributes such as type, availability, or load.

The final layer in the Grid architecture comprises the user applications. Applications are constructed in terms of services defined at each layer in the Grid structure. At each layer, well-defined protocols provide access to some useful service: resource management, data access, resource discovery, and so forth. At each layer, protocols and services are used to perform desired actions.

B. Globus Toolkit

The Globus Project [20] provides software tools that make it easier to build computational grids and grid-based applications. These tools are collectively called The Globus Toolkit. The Globus Toolkit is used by many organizations to build computational grids that can support their applications. Globus has a layered architecture that includes the *fabric*, *connectivity*, *and resource*, *collective* and *application* layers, as in Figure 2.

The fabric layer defines the control of local "things" that mix a diverse set of sharable resources. The connectivity layer is the core layer of Globus, which includes the Internet protocol, DNS, etc. It provides a uniform authentication, authorization, and message protection mechanism (Grid Security Infrastructure, abbreviated as GSI) in a multiple institutional setting. The core layer also includes the resource layer.

The resource layer involves protocols and services such as Grid Resource Allocation Management (GRAM), high-performance data access and transport (GridFTP), access to structure and state information (Grid Resource Information Service, GRIS) and other emerging services such as accounting. The collective layer provides the indexing service called *Meta directory service* (*MDS*), which allows viewing on dynamic resource collections assembled by a community. It also involves replica management and location, etc. Finally, the application layer is an application developed by programmers and invoking interfaces provided by various layers.

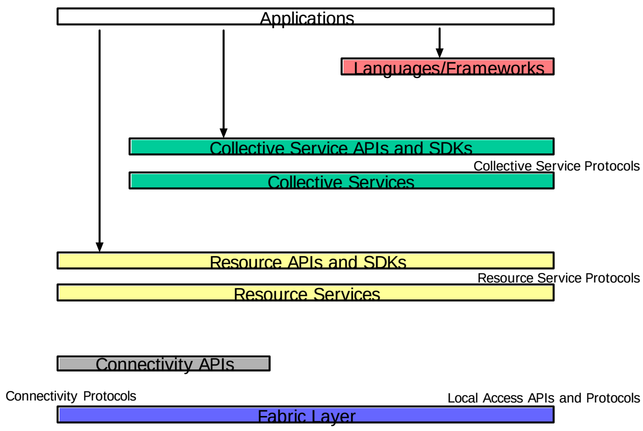


Fig. 2. The Globus Grid architecture: protocols, services, and APIs at each level.

The composition of the Globus Toolkit can be pictured as three pillars: Resource Management, Information Services, and Data Management [20]. Each pillar represents a primary component of the Globus Toolkit and makes use of a common foundation of security. GRAM implements a resource management protocol, MDS implements an information services protocol, and GridFTP implements a data transfer protocol. They all use the GSI security protocol at the connection layer.

GRAM is designed to provide a single common protocol and API for requesting and using remote system resources, by providing a uniform, flexible interface to, local job scheduling systems, while MDS is designed to provide a standard mechanism for publishing and discovering resource status and configuration information. At last, GridFTP is a high-performance, secure, reliable data transfer protocol optimized for high-bandwidth wide-area networks.

C. MPICH-G2

MPICH-G2 [17, 21-23] is a grid-enabled implementation of the message-passing interface for grid environments. It allows you to interconnect a number of computational resources, potentially of different architectures, to run your MPI applications. MPICH-G2 automatically converts data in messages sent between computing systems of different architectures and supports multi-protocol communication. Most of existing parallel programs written with MPI are able to be executed in the Globus-based infrastructure just after recompilation [16].

III. PROVIDENCE UNIVERSITY'S COLLEGE OF COMPUTING AND INFORMATICS GRID COMPUTING PLATFORM

In this section, we introduce the grid-computing platform, constructed by interconnecting a number of cluster platforms, currently installed in different laboratories and used for different topics of research, distributed across different floors inside Providence University's College of Computing and Informatics. In next subsections, we introduce application tools and parallel software applications built on top of this grid platform and available for our community. These tools and applications have been developed as topics of research conducted by our teams.

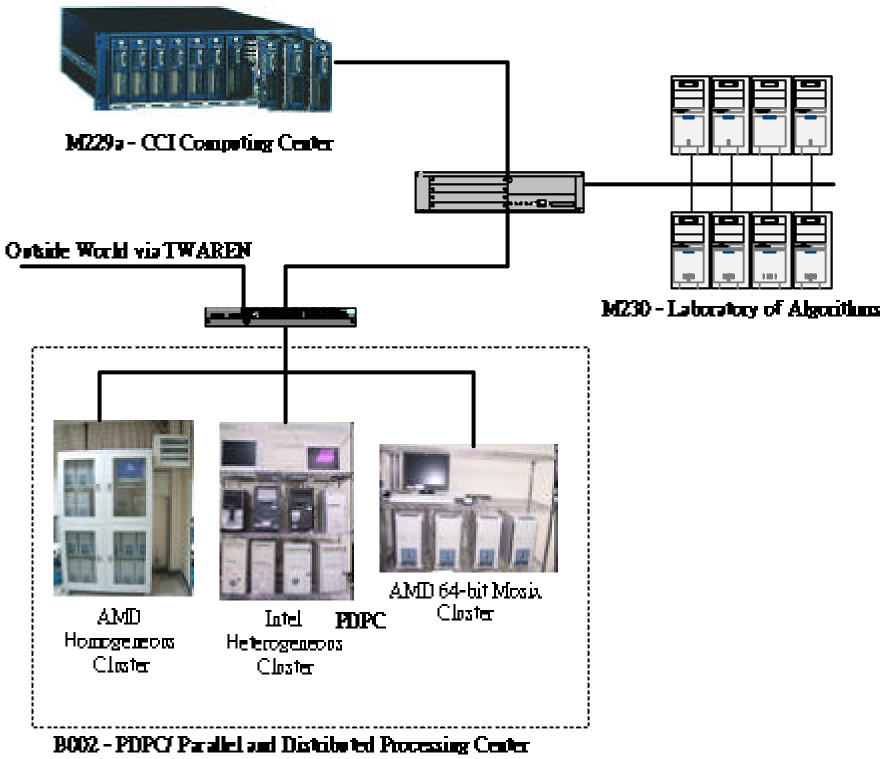
A. Grid Platform Hardware

The PCGrid grid-computing platform, standing for Providence University's College of Computing and Informatics grid platform, consists of five cluster platforms located in different floors inside the College of Computing and Informatics (CCI). The project of constructing such grid infrastructure is aimed to increase Providence University's computational resources and sharing these resources among investigators and researchers in fields such as parallel compilers, parallel software, data distribution, bioinformatics, performance analysis and visualization toolkit, parallel application graph representation, computing node selection, thread migration and scheduling in grid environments, among others.

The grid platform formed by interconnecting five cluster platforms via Gigabit Ethernet (1Gb/s). The PCGrid platform has a total number of 45 computing nodes, of different CPU speed and memory sizes, and a total storage of more than 3TB. The detail hardware specification of each cluster platform is listed in Table 1, while Figure 3 illustrates the PCGrid computing infrastructure.

Table 1. PCGrid grid platformhardware specifications.

Site	Qty (# nodes)	CPU / Memory	Local Network
AMD Cluster	17	1 frontend + 16 nodes: AMD Athlon 2400+, 1GB DDR memory	Gigabit Ethernet
Intel Cluster	9	Computing nodes with different CPUs (varies from 350MHz to 2.4GHz) and memory sizes (varies from 64MB to 1GB memory)	Fast Ethernet
AMD64 Cluster	4	AMD Sempron 64-bit 2800+ CPU, 1GB memory	Gigabit Ethernet
Intel-P4 Cluster	9	1 frontend (Intel P4 3GHz CPU, 2GB) + 8 nodes (Intel P4 3GHz CPU, 1GB memory)	Gigabit Ethernet
IBM Blade Server	6	Each blade contains 2 PC950 CPUs, 1GB memory	High speed bus



M229a - CCI Computing Center
Outside World via TWAREN
M230 - Laboratory of Algorithms
AMD
Homogeneous
Cluster
Intel
Heterogeneous
Cluster
PDPC
AMD 64-bit Mosix
Cluster
B002 - PDPC/ Parallel and Distributed Processing Center

IV. Researches Under Investigation

It places special emphasis on the high performance requirements of applications developed on the Grid. This means that the integration of systems and resources, as well as heterogeneity and dynamic situations management, must explicitly handle the case of Grid computing nodes (which in general are geographically distributed or located on private virtual networks) as high performance systems, such as parallel machine architectures or clusters.

Researches extend to all the platform levels, from high bandwidth networks to middleware services and, in particular, to resource management as well as tools and programming environments as Figure 4. In this section, we will describe a number of ongoing projects being developed by our research teams.

Bioinformatics	Programming Environments
MPI Performance Evaluation toolkit	
Resource Broker	Data Replication
Security	
Middleware Multicast and SGM	
High Speed Networks	

Fig. 4. Ongoing projects using PCGrid grid platform.

A. *Middleware*

At the middleware level, research on resource management includes aspects of maximum importance such as discovery, brokering, scheduling, monitoring and performance evaluation/prediction [26-28]. In addition to aspects concerning programming environments and resource management, the software technology studied in Grid. It includes some fundamental aspects related to middleware (also see Figure 6):

- security: secure Grid environments and cooperation among Grid environments belonging to different organizations,
- data intensive services: federated database services, visualization and hierarchical management of data and meta-data according to advances and high-performance techniques ,
- knowledge discovery services: Grid services (data mining, search engines, etc.) that provide consistent, efficient and pervasive access to high end computational resources,
- Grid portals: Grid enabled application services, e.g., providing the user with the possibility to submit tasks to remote jobs and collect results via Web interfaces,
- The design and implementation of scientific libraries that can be used in a heterogeneous and dynamic context, such as the Grid, completes the research on programming environments.

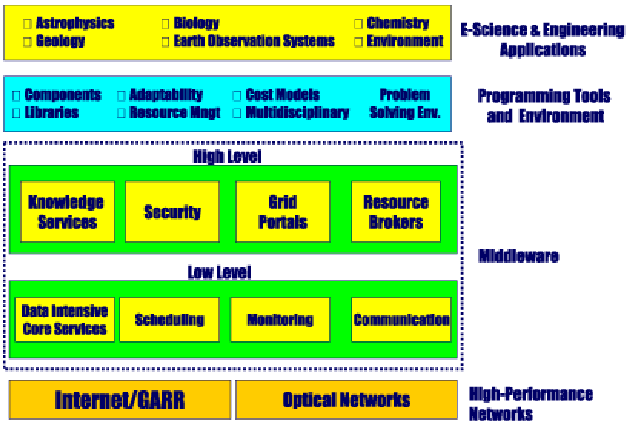


Fig. 5. Grid platform middleware issues.

B. *Multicast and SGM*

Multicast is an important technique for optimizing bandwidth for a wide range of applications, especially digital multimedia, by allowing for the total number of data streams to be managed more efficiently. For example, by using multicast techniques, it is possible to direct streams only to nodes where they are required and can reduce the total number of streams required in network segments by avoiding duplication. While today's multicast schemes are scaleable in the sense that they can support very large multicast groups, these schemes have problems when a network needs to support a very large number of distinct multicast groups. The Small Group Multicast (SGM) project centers on a new approach to multicast that complements today's multicast schemes.

C. *High Speed Networks*

The study of the high speed networks needed to support enabling Grid platforms for scalable VOs is an internationally recognized 'hot' topic. Within this research activity, an important role is played by experiments on very high bandwidth optical networks, based on photonic technology for Grid platforms with high performance sites that extend to the metropolitan area.

D. *Security*

In this study, we focus on the access control issue in sharing mechanism. The service present in PCGrid platform consists of heterogeneous administrative domains, in which the access control mechanism and security policy are not uniform. The purpose and need of access control are also different in various organizations. Access control is one of the key issues in the study of resource sharing in Grid computing. Information sharing requests authorization across the boundaries of organizations and the sharing relations are under control, that is to say, which resources will be shared to someone should be defined definitely. At the same time sharing mechanism can not force local systems to change their security mechanism in substance .The principle of autonomous control requires that each domain should specify its access policy and keep in the final control of the resource. Grid technology put emphasis upon versatile and open standard and decentralized control.

We will integrate multiple computational resources located in different floors of PU/CCI and to construct a high-performance and secure Computational Grid and Data Grid environment for data streaming and services. In a Computational Grid and Data Grid environments that users are authorized and classified into different privilege classes to form a hierarchical structure, a user belonging to higher-privileged class, and which will have access right to messages created or owned by users in a lower-privileged class; while the opposite is not allowed. To meet the access control requirement in a user hierarchy and the special characteristics of Grid computing, must propose a dynamic key management scheme suitable for hierarchical control systems and that appeared in computer communications. It is useful in solving the key management problem for a privileged hierarchy using related parameters. We intend to propose a new access control method base on the elliptic curves to overcome the exist problems, and furthermore to strengthen the security level.

E. Bioinformatics

The critical issues of bioinformatics are to discover the secret information under the DNA code, to understand how the biology system work, and to improve the development of medical, health and environment. Due to the vest biology data and updating rate, the greatest challenge of bioinformatics research is the demand for computing resource. Grid computing technology provides the most economical high performance computing resource. Through the Internet, Grid can extend its computing power unlimited, and feasible for bioinformatics.

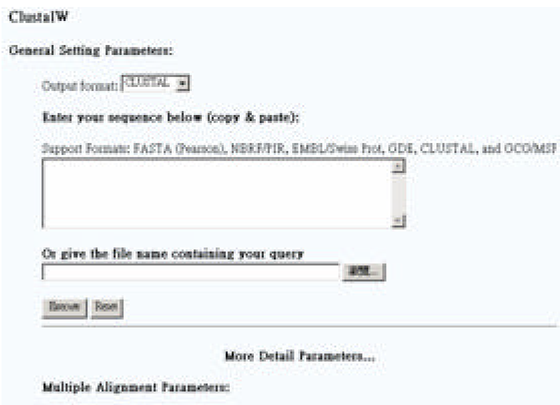


Fig. 6A. The web interface of ClustalW.

Among many bioinformatics research issues, we focus on three most important problems: biology database searching, multiple sequences alignment and evolution tree construction. We construct ClustalW-MPI [29, 30] on our PCGrid, and use parallel technology to improve the computation of multiple sequences alignment and evolution tree; meanwhile, we also include a series of BLAST [31] tools to enhance sequence and database searching. All these system have a common web user interface for usability. For examples, Figure 6A and 6B is the screenshot of ClustalW and Bl2seq.

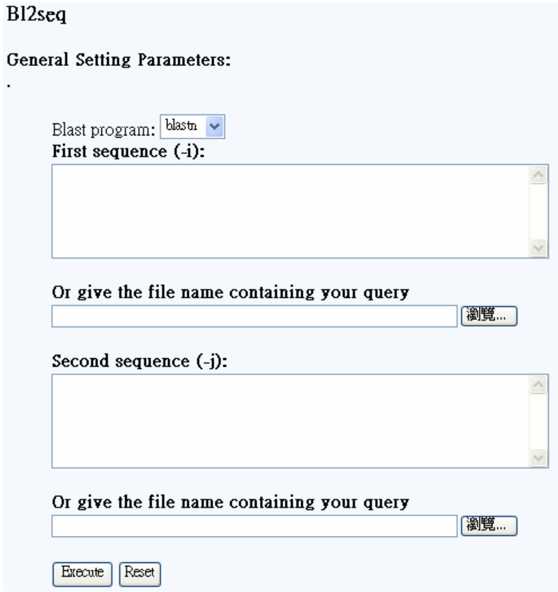


Fig. 6B. The web interface of Bl2seq.

F. Grid Economy

The goal of Grid Economy research is to develop an efficient and economical Cost Mode to advance the usability and management of Grid resources. We integrated Technology Cost [32] and Deadline Scheduling [33, 34] in our Cost Model, and build a web interface for user easy to setup their requests: budget, deadline of run time, program upload and computing resource selection. Figure 7A and 7B is the screenshot of user interface and process result.

Differ from the usual economy model that only the provider can define their cost, we involve user's demand to Technology Cost to improve the cost policy and gain the price balance. Besides the budget, the deadline of processing can be a user's demand in our system, we use Deadline Scheduling Algorithm to minimized the number of unfinished job before deadline.

Fig. 7A. The interface for user to upload program and setup their budget and deadline.

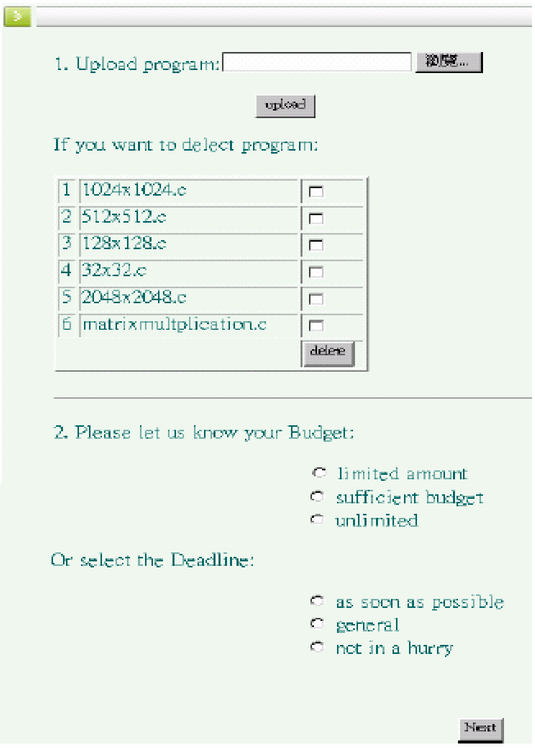


Fig. 7B. Process result.

G. Resource Broker

The resource broker research conducted was to develop an interface that its main function is to match the available computational resources upon user's requests. The use of the resource broker provides a uniform interface to access any of the available and appropriate resources using user's credentials. The resource broker runs on top of the Globus Toolkit. Therefore, it provides security and current information about the available resources and serves as a link to the diverse systems available in the grid platform.

We implement a grid resource broker that considers the network bandwidth and latency for loosely/tightly coupled applications, cluster MDS issues and application chrematistics. Two types of resource broker are being conducted in our TIGER Grid testbed: manual and automatic computing node manual selection.

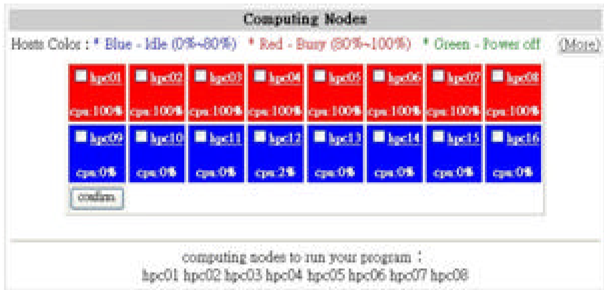


Fig. 8A. Computing Node manual selection simple mode.

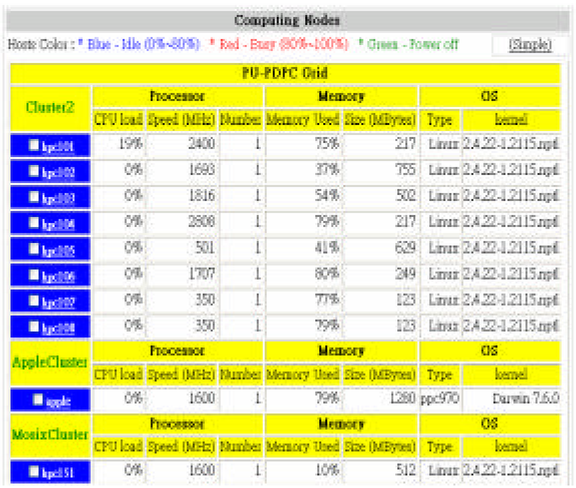


Fig. 8B. Real-time display of all computing nodes status in complete mode.

The first one shows each computing node's real-time colored status, being the color 'red' as CPU utilization of 80%~100%, color 'blue' when CPU utilization is 0%~80%, and color 'green' when the computing node is off. It had two modes: simple and complete. The complete mode of the broker show, in real-time basis, complete information of each computing node, such as memory and CPU utilization. In addition, it shows also OS and kernel information. The simple and complete mode's snapshots are illustrated in Figures 8A and 8B, respectively.

H. Process Monitoring

It is possible to visualize the system status of all computing nodes of grid. The webpage shows system information under the title "System Status". Moreover, you can press "Refresh Data" button to see recent system status, as in Figure 9.

System Status (Refresh Data)

HOST	PID	USER	PRI	NI	SIZE	RSS	SHARE	STAT	%CPU	%MEM	TIME	CPU	COMMAND
hpc105	8312	pdpc	17	0	1116	1116	860	R	2.9	0.1	0:00	0	top
hpc108	28688	pdpc	17	0	1160	1160	860	R	1.9	0.9	0:00	0	top
hpc107	20505	pdpc	16	0	1116	1116	860	R	1.9	0.8	0:00	0	top
hpc101	28725	pdpc	17	0	1104	1104	772	R	1.9	0.4	0:00	0	top
hpc105	8314	pdpc	24	0	524	524	436	S	0.9	0.0	0:00	0	awk
hpc101	28812	root	15	0	0	0	0	SW	0.9	0.0	1:54	0	nfsd
hpc106	2931	gdm	15	0	9232	9232	7248	S	0.0	3.6	0:00	0	gdmgreeter
hpc106	2780	root	15	0	90448	8028	2116	S	0.0	3.1	0:00	0	X
hpc107	2052	gdm	16	0	4028	2760	2308	S	0.0	2.1	0:02	0	gdmgreeter
hpc108	10186	gdm	15	0	3764	2480	2160	S	0.0	1.9	0:02	0	gdmgreeter
hpc101	3198	ntp	16	0	3844	3844	2204	S	0.0	1.7	0:00	0	ntpd
hpc103	23366	gdm	15	0	9232	9232	7248	S	0.0	1.7	0:00	0	gdmgreeter
hpc104	3639	gdm	15	0	4648	3388	2676	S	0.0	1.5	0:00	0	gdmgreeter
hpc101	19477	gdm	15	0	4684	3344	2624	S	0.0	1.4	0:01	0	gdmgreeter
hpc105	18690	gdm	15	0	9240	9240	7248	S	0.0	1.4	0:01	0	gdmgreeter
hpc106	2286	xfs	16	0	3492	3492	1040	S	0.0	1.3	0:00	0	xfs
hpc105	1956	root	15	0	17144	8692	2152	S	0.0	1.3	0:03	0	X
hpc106	2244	root	16	0	3076	3076	2940	S	0.0	1.2	0:00	0	httpd
hpc102	2513	root	15	0	41784	8524	2148	S	0.0	1.1	0:02	0	X
hpc102	23590	gdm	15	0	9232	9232	7248	S	0.0	1.1	0:00	0	gdmgreeter

Fig. 9. The webpage that lists system status of all computing nodes of PCGrid grid platform.

I. MPI Performance Evaluation toolkit

In this research, we design and implement a MPI parallel program performance monitoring and analysis toolkit for PCGrid computing environment, by providing graphical performance data visualization when executing a MPI parallel application in such environment. This toolkit is extremely useful for the development of a MPI application, show performance data of each computing node involved (see Figure 10A), to tune its performance, and to perform "what-if" analysis, by comparing different versions during the development of such application (see Figure 10B).

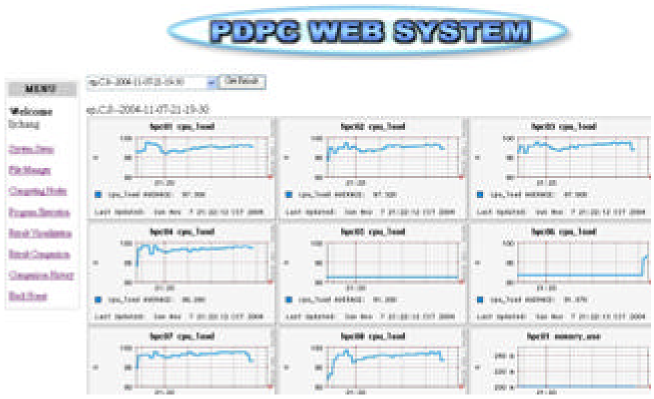


Fig. 10A. Performance data of each computing node involved in computation of PCGrid platform.

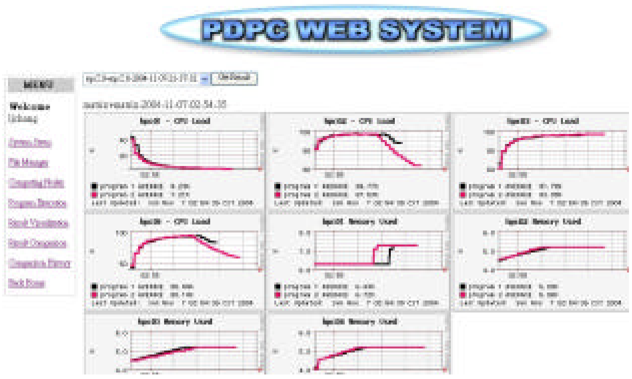


Fig. 10B. Performance comparison of two execution results, computing node by computing node, CPU load and memory usage.

J. Data Replication

The term "Data Grid" aggregate distributed resources to produce results for large size problems. Most of these Data Grid applications are executed simultaneously and access a large number of shared data files in grid environment. In certain data intensive scientific applications, such as high-energy physics, bioinformatics applications and astrophysical virtual observatory, we confront with huge amount of data.

A Data Grid provides two essential basic services, which are a secure, reliable, efficient data transport protocol and replica management. The high-speed transport protocol, GridFTP, extends the popular FTP protocol with some new features required for Data Grid applications, such as partial file transfer and third-party transfer. The replica management service take advantage of replica catalog with GridFTP transfer to provide for the creation, registration, location and management of data replicas.

As our PCGrid research, we propose a cost model according to the three significant parameters: network bandwidth, CPU load and I/O state. Although the network situation is constantly changing and the storage equipments are busy or idle, we are investigating in a cost model to determine the best replica immediately, because our cost model is based on the system monitoring information that update continuously.

K. *Programming Environments*

Programming tools and environment level the high performance requirement implies that, when designing scalable virtual organizations (VOs), a unifying approach should be used. This approach takes into account both aspects related to the distribution of computations and resources, and those related to parallelism. The programming environment must therefore be characterized by a high degree of portability on different hardware-software systems (or different hardware-software combinations) in a heterogeneous and dynamic context. Portability must be guaranteed for not only code but should also ensure that the performance matches the configuration of the target system at hand. Tool interoperability and high performance application reuse are also fundamental to the programming environment.

V. CONCLUSION AND FUTURE WORK

The PCGrid grid-computing platform presents a unique set of challenges and opportunities with regard to investigations in Grid Computing. All ongoing projects are equally important to advance and support the development of such dynamic computing environment. We are currently continuing our investigations to advance the inter-operability of PCGrid grid computing platform. Our teams have begun to deploy Globus Toolkit 4, as a way to increase the reliability, portability and performance of our computing platform. In addition, we have started to develop researches in security, to support and handle the stringent security requirements needed for grid platform.

Finally, we could show in this research paper possibility of increasing computational power, sharing idle computational resources and knowledge sharing when building up a grid infrastructure. We could show the viability of constructing such platform inside a college, extending in future to interconnect computational resources in whole campus, with the goal of increasing research interaction and cooperation among faculties inside a university.

REFERENCES

[1] B. Allcock, J. Bester, J. Bresnahan, A. L. Chervenak, I. Foster, C. Kesselman, S. Meder, V. Nefedova, D. Quesnal, and S. Tuecke. "Data Management and Transfer in High Performance Computational Grid Environments," *Parallel Computing Journal*, Vol. 28 (5), May 2002, pp. 749-771.

[2] B. Allcock, J. Bester, J. Bresnahan, I. Foster, J. Gawor, J. A. Insley, J. M. Link, and M. E. Papka. "GridMapper: A Tool for Visualizing the Behavior of Large-Scale Distributed Systems," In *Proceedings of 11th IEEE International Symposium on High Performance Distributed Computing (HPDC-11)*, July 2002.

[3] M. Baker, R. Buyaa, D. Laforenza, *Grid and Grid Technologies for Wide-Area Distributed Computing*, available at http://www.csse.monash.edu.au/~rajkumar/papers/gridtech.pdf

[4] F. Berman, A. Chien, K. Cooper, J. Dongarra, I. Foster, D. Gannon, L. Johnson, K. Kennedy, C. Kesselman, J. Mellor-Crummey, D. Reed, L. Torczon, and R. Wolski. "The GrADS Project: Software Support for High-Level Grid Application Development," *International Journal of High-Performance Computing Applications*, 15(4), 2002.

[5] M. Chetty, R. Buyya, Weaving computational Grids: How analogous are they with electrical Grids? *Journal of Computing in Science and Engineering* (CiSE) 2001; (July - August).

[6] K. Czajkowski, I. Foster, and C. Kesselman. "Resource Co-Allocation in Computational Grids," In *Proceedings of the Eighth IEEE International Symposium on High Performance Distributed Computing (HPDC-8)*, pp. 219-228, 1999.

[7] K. Czajkowski, S. Fitzgerald, I. Foster, and C. Kesselman. "Grid Information Services for Distributed Resource Sharing" In *Proceedings of the Tenth IEEE International Symposium on High-Performance Distributed Computing (HPDC-10)*, August 2001.

[8] A. Chervenak, E. Deelman, I. Foster, L. Guy, W. Hoschek, A. Iamnitchi, C. Kesselman, P. Kunst, M. Ripeanu, B, Schwartzkopf, H, Stockinger, K. Stockinger, and B. Tierney. "Giggle: A Framework for Constructing Sclable Replica Location Services," In *Proceedings of Supercomputing 2002 (SC2002)*, November 2002.

[9] I. Foster, C. Kesselman, eds., *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, San Francisco, Calif. (1999).

[10] I. Foster et al., *The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration*, available at http://www.globus.org/research/papers/ogsa.pdf

[11] I. Foster, *What is Grid? A Three Point Checklist*, available at http://www.gridtoday.com/02/0722/100136.html

[12] I. Foster, *The Grid: A New Infrastructure for 21st Century Science*, available at http://www.gridtoday.com/02/0805/100191.html

[13] I. Foster, *Internet Computing and the Emerging Grid*, available at http://www.nature.com/nature/webmatters/grid/grid.html

[14] I. Foster, C. Kesselman, and S. Tuecke. "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," *International Journal of Supercomputer Applications*, 15(3), 2001. Available at http://www.globus.org/research/papers/anatomy.pdf

[15] I. Foster, C. Kesselman, G. Tsudik, and S. Tuecke. "A Security Architecture for Computational Grids," In *Proceedings of 5th ACM Conference on Computer and Communications Security Conference*, pp. 83-92, 1998.

[16] M. Ripeanu, I. Foster and A. Iamnitchi. "Mapping the Gnutella Network: Properties of Large-Scale Peer-to-Peer Systems and Implications for System Design," *IEEE Internet Computing*, vol 6(1), January-February 2002.

[17] A. Roy, I. Foster, W. Gropp, N. Karonis, V. Sander, and B. Toonen. "MPICH-GQ: Quality-of-Service for Message Passing Programs," In *Proceedings of the IEEE/ACM Supercomputing Conference (SC2000)*, November 4-10, 2000.

[18] V. Welch, F. Siebenlist, I. Foster, J. Bresnahan, K. Cajkowski, J. Gawor, C. Kesselman, S. Meder, L. Pearlman, and S. Tuecke. "Security for Grid Services," In *Proceedings of IEEE Twelfth International Symposium on High Performance Distributed Computing (HPDC-12)*, June 2003.

[19] Global Grid Forum, http://www.ggf.org/

[20] The Globus Project, http://www.globus.org/

[21] MPI Forum, http://www.mpi-forum.org/

[22] MPICH, http://www-unix.mcs.anl.gov/mpi/mpich/

[23] MPICH-G2, http://www.hpclab.niu.edu/mpi/

[24] A. Beguelin and J. Dongarra, Solving computational Grand

Challenges using a network of heterogeneous supercomputers, Proceedings of 5th SIAM Conference on Parallel Processing, 1991.

[25] W. Gropp, E. Lusk, and A. Skjellum, Using MPI: portable parallel programming with the Message Passing Interface, MPI Press, Cambridge, MA, 1994.

[26] K.C. Li, C.T. Yang, L.J. Chang, H.Y. Cheng, H.C. Chang, L.T. Lee, "Implementation of Visuel MPI Parallel Program Performance Analysis Tool for Cluster Environments", in AINA'2005 The First International Workshop on Information Networking and Applications, IEEE CS Press, Taipei, Taiwan, 2005.

[27] B.P. Miller et al. The Paradyn parallel performance measurement tools, *IEEE Computer*, Special issue on performance evaluation tools for parallel and distributed computer systems, 28, 11, pp.37-46, 1995.

[28] C.T. Yang, C.L. Lai, P.C. Shih, and K.C. Li, "Resource Broker for Computing Nodes Selection in Grid Computing Environments", in GCC' 2004 The 3rd International Conference on Grid and Cooperative Computing, LNCS 3251 Springer-Verlag Heidelberg, H. Jin, Y. Pan, N. Xiao, J. Sun (Eds.), Wuhan, China, 2004.

[29] Thompson, J.D., Higgins, D.G. and Gibson, T.J. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice. Nucleic Acids Research, 22:4673-4680.

[30] Li KB. ClustalW-MPI: ClustalW analysis using distributed and parallel computing.

[31] Altschul, Stephen F., Thomas L. Madden, Alejandro A. Schfer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997), "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs", Nucleic Acids Res. 25:3389-3402. Bioinformatics. 2003 Aug 12;19(12):1585-6.

[32] Zhengyou Liang, Ling Zhang, Shoubin Dong and Wenguo Wei, "Charging and Accounting for Grid Computing System," Springer-Verlag Berlin Heidelberg, pp. 644-651, 2004

[33] Atsuko Takefusa, Henri Casanova and Satoshi Matsuoka, and Francine Berman, "A Study of Deadline Scheduling for Client-Server Systems on the Computational Grid," 10th IEEE International Symposium on High Performance Distributed Computing （HPDC-10'01） , San Francisco, California, pp. 406-415, August 07-09 2001.

[34] Buyya R, Murshed M. and Abramson D., "A Deadline and Budget Constrained Cost-Time Optimization Algorithm for Scheduling Task Farming Applications on Global Grids," The 2002 International Conference on Parallel and Distributed Processing Techniques and Applications, Las Vegas, Nevada, USA, June 2002.