Enabling Automated Deployment for Cluster Computing in Distributed PC Classrooms

Shuen-Tai Wang, Ying-Chuan Chen, Hsi-Ya Chang

***Abstract*—**The rapid improvement of the microprocessor and network has made it possible for the PC cluster to compete with conventional supercomputers. Lots of high throughput type of applications can be satisfied by using the current desktop PCs, especially for those in PC classrooms, and leave the supercomputers for the demands from large scale high performance parallel computations. This paper presents our development on enabling an automated deployment mechanism for cluster computing to utilize the computing power of PCs such as reside in PC classroom. After well deployment, these PCs can be transformed into a pre-configured cluster computing resource immediately without touching the existing education/training environment installed on these PCs. Thus, the training activities will not be affected by this additional activity to harvest idle computing cycles. The time and manpower required to build and manage a computing platform in geographically distributed PC classrooms also can be reduced by this development.

***Keywords*—**PC cluster, automated deployment, cluster computing, PC classroom.

Open Science Index, Computer and Information Engineering Vol:6, No:5, 2012 publications.waset.org/1719.pdf

I. Introduction

CLUSTER computing [1] is a topic of growing importance, due to its role in massively-parallel supercomputing applications, grid computing, cloud computing, and other applications where large amounts of data have to be managed and manipulated immediately. In essence, PC cluster [2,3] is the major fundamental platform for doing cluster computing. By the rapid improvement of the microprocessor and network, it had made possible for the PC cluster to compete with conventional supercomputers.

A PC cluster is a type of parallel or distributed processing system that consists of a collection of interconnected stand-alone PCs working together as a single, integrated computing resource. In fact many powerful supercomputers currently in use are made of microprocessors and which usually are even a generation behind the fastest processors used in current desktop PC. PC cluster also provides an inexpensive computing resource to educational institutions or research organizations. They need not invest millions of dollars to buy parallel computers for the purpose of running parallel computations.

Nowadays, many high throughput types of applications can be satisfied by using the desktop PCs, especially for those in PC classroom [4]. We can consider that there should be a place such like PC classroom where exists many PCs can be integrated as a PC cluster to do many computations.

So the PC classroom is an unheeded resource or underutilized in academic institutes and colleges, but it is a potential computing resource to support some computations.

The PCs in PC classroom are usually setup during the daytime for education and training purposes and then shut down at night. One interesting topic can be raised is how to exploit the computing resource of the power-off PCs, and without touching the existing education/training environment installed on these PCs. Thus, the training activities will not be affected by this additional activity to harvest idle computing cycles.

To do so, in this paper, we present an innovative way and automated mechanism to solve the idling issue by deploying pre-configured cluster computing environment that can utilize the existing computing power residing in PC classrooms. We have implemented it as a package call "Classroom Cluster". Classroom Cluster can transform the PCs into cluster computing resource immediately that can be used when the PCs are normally not in use without touching the existing education/training environment. This can be achieved by "Root over NFS" diskless environment. So it does not touch the client hard drive, therefore, existing operating systems, along with all the software and applications installed on them, are preserved. In addition to automation and manageability, many middleware are packaged, such as resource management system, dynamic power management, accounting management, monitoring tools, and the Message Passing Interface (MPI) [5] libraries for parallel program. On the other hand, we also modified some Grid-Related middleware for collaborating the geographically distributed PC classrooms to perform reliable and efficient sharing of computing resources, and provide single entry interface for users for submitting, monitoring and controlling jobs in distributed environment.

This paper presents an effective way to obtain the existing computing power residing in PC classrooms for cluster computing. It works well especially for computationally bound applications. The rest of this paper is organized as follows. Section 2 lists the related works. Section 3 gives a description of hardware/software architecture. Section 4 gives some details of the implementation. Performance evaluation and analysis will be presented in section 5. Finally, section 6 presents the conclusion and future work.

II. Related Work

A. Rocks

The Rocks [6] toolkit is a full cluster package distribution that uses Red Hat kickstart to install the computing nodes. It can take a fresh perspective on management and installation of clusters to dramatically simplify software version tracking, and cluster integration. Full automation of the PC cluster installation is its major advantage.

But for the PCs in PC classroom, by deploying Rocks directly will format all the hard drives. Furthermore, considering these packages will provide more quickly image backup/restore functions, but the installation time of real system may not be efficient for such situation of PC classroom.

S.T. Wang is with the National Center for High-Performance Computing, Taiwan, R.O.C. (e-mail: stwang@nchc.org.tw).

Y.C. Chen is with the National Center for High-Performance Computing, Taiwan, R.O.C. (e-mail: ycc0301@nchc.org.tw).

H.Y. Chang is with the National Center for High-Performance Computing, Taiwan, R.O.C. (e-mail: jerry@nchc.narl.org.tw).

Open Science Index, Computer and Information Engineering Vol:6, No:5, 2012 publications.waset.org/1719.pdf

B. SCE

The SCE (Scalable Cluster Environment) project was developed at Kasetsart University in Thailand [7]. SCE is a software suite that can lead user efficient to build PC cluster. SCE includes tools to install, monitor and manage computing nodes, and a batch queuing system and scheduler to address the difficulties in deploying and maintaining clusters. But unlike Rocks, SCE can't provide an entire, self-contained, cluster-aware installation. This may lead to users very difficult to add and customize cluster functionality. In common with Rocks, the installation time of real system may not be efficient for such situation of PC classroom.

C. OSCAR

OSCAR [8] is a collection of common clustering software tools in the form of tar files that are installed on top of a Linux distribution on the frontend computer. When integrating computing nodes, IBM's Linux Utility for cluster Install (LUI) operates in a similar manner to Red Hat's Kickstart. The drawbacks of OSCAR is it requires a deep understanding of cluster architectures and systems, relies upon a 3rd-party installation program, and has fewer supported cluster-specific software packages than Rocks and SCE. It also will format all the hard drives of PCs, so it may not suitable for PC classroom.

D. ClassCloud

ClassCloud [9] is designed for building Cloud infrastructure in PC classrooms. ClassCloud can switch the PCs into virtual platform, and it also focuses on investigation of PC classroom. The goal of ClassCloud is to build up a basic cloud computing environment for testing. It uses Xen [10] to construct a virtual computing resource. Although adopting virtual machine delivers high flexibility, it has a native performance issue, and it is mainly designed for cloud environment and may not meet the requirements of gathering the idle computing power in PC classroom.

III. System Architecture

Figure 1 shows the hardware architecture of Classroom Cluster. It consists of a head node, backup node and numerous computing nodes. The head node has two Ethernet interfaces. One is assigned a public IP address and connected to internet. The other is connected to the local classroom switch. Both head node and computing node are installed with the same operating system to avoid the problem of missing library files.

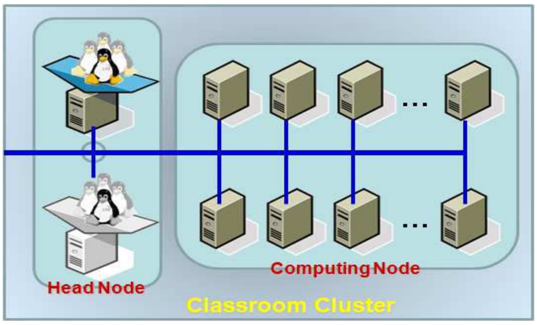

Fig. 1 Hardware architecture

Figure 2 shows the software stack of Classroom Cluster. It can be divided into two major modules: Diskless and Kernel. First, the Kernel module depends on the proper setups of diskless environment. The Diskless module includes: TFTP, NFS, NIS, DHCP and PXE/etherboot [11] services. Above the diskless layer, several software are packaged into the Kernel module for the system, such as local batch queue system, accounting software, monitoring tools, power management module and MPI libraries. Most of the software are widely accepted and are tailored and tuned for helping automation and manageability. We also wrapped the pre-configured software into a package for some Linux distributions, including CentOS, Fedora, and Red Hat. With this one-size-fits-all solution staged, we can generally eliminate the efforts of building Classroom Cluster.

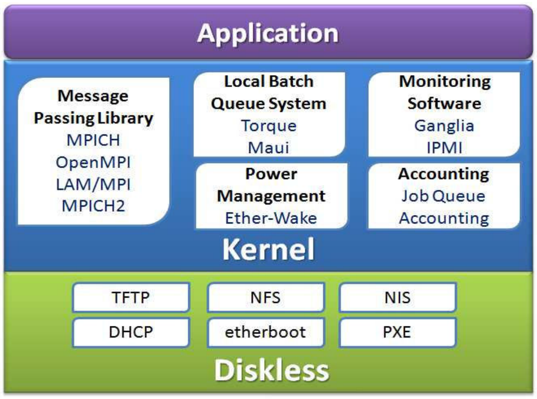

Fig. 2 Software stack

Figure 3 shows the experimental environment of our development. Now our organization – National Center for High-performance Computing (NCHC) [12] has three business units located at three science parks in Taiwan, and each business unit has two PC classrooms. We built the experimental environment that consists of five Classroom Clusters which are widely distributed in PC classrooms over Taiwan. The front node is a login node located in the top of system for users to provide a single entry interface for submitting, monitoring and controlling jobs.

After well deployment, all PCs in the PC classrooms could be scheduled. Users can login the front node and submit their jobs to resource broker anytime. The resource broker will fetch the jobs and send to the applicable Classroom Cluster by its scheduling policy. And then even when the work horses of the Classroom Cluster is not available at the moment. The submitted jobs will be queued and wait for computing resources to become available. When the PCs are available, the system will fetch the suitable jobs, parse the requirements, and remote power on exact number of the PCs. After jobs are completed, the outputs will be sent back to the front node.

Open Science Index, Computer and Information Engineering Vol:6, No:5, 2012 publications.waset.org/1719.pdf

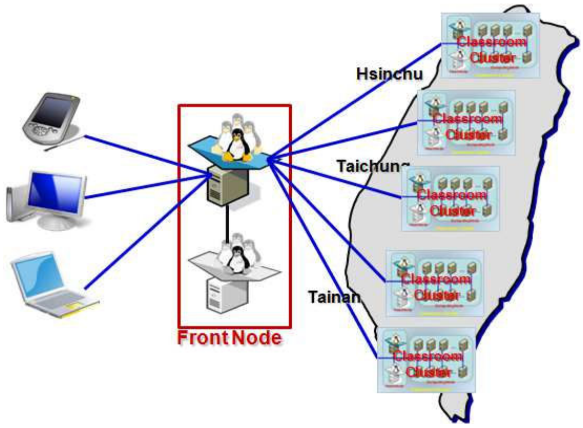

Fig. 3 Experimental environment

IV. Implementation

A. Diskless Environment

The diskless is a PC without disk drives, which employs network remote booting to load its operating system from a server. We implemented such mechanism to managing the deployment of the Linux operating system across many compute nodes for the Classroom Cluster. It uses PXE/etherboot, NFS, and NIS to provide services to PCs so that it is not necessary to install operating system on the PC's hard drives individually, and so on, it does not touch the hard drives, therefore, the original operating system installed on the PCs will be unaffected. During the deployment of diskless environment, the administrator needs to power on PCs one by one and gather their MAC addresses. The diskless module will collect MAC addresses and generate the associated configuration file for DHCP and PXE/etherboot services. Furthermore, the list of MAC addresses will be the target of power management mentioned in the following section. After diskless environment being valid, the head node will become a powerful server, which provides initial RAM disk, IP address for PXE boot, NFS and NIS service for a PC classroom.

B. Power Management

In recent years the way how we use PC cluster might also increase energy burden. We developed a new approach to reduce energy utilization in PC classroom for Classroom Cluster. We do this work on the integration of resource management system and remote power management system that aims at reducing power consumption such that they suffice for meeting the minimizing quality of service. In particular, our approach relies on recalling services dynamically onto appropriate amount of the PCs according to user's computing job request and temporarily shutting down the computers after finish in order to conserve energy. As shown in Figure 4, users can submit their jobs to Classroom Cluster anytime, even if the cluster is not available at the time the job is submitted. The submitted jobs are queued and then wait for the computing resources to become available, typically at night time when the classroom PCs are free from their assigned daytime duty. When the PCs become available, the head node of Classroom Cluster fetches the applicable jobs, parses the requirements, and remotely powers on the correct number of PCs by Wake-on-LAN [13] protocol. After the job was completed, Classroom Cluster will power the PCs down. Our implementation currently relies on checking the local queuing system (i.e. Torque [14]) job pool and then decides to shut down which compute nodes when no new job was submitted. By powering off idle PCs, it can significantly save more energy than always keeping all PCs running.

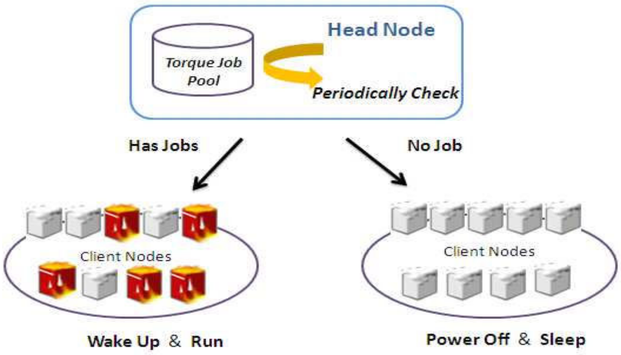

Fig. 4 Scenario of power management

C. Multi-Cluster Cooperator

After well deployment of Classroom Cluster in PC classroom, we hope that these distributed Classroom Clusters can collaborate and form as a whole resources for users. So we refer to Grid [15] architecture to achieve this work. Unlike the complexity of standard Grid middleware, the front and head nodes are all in our administrative domain, adopted the same operating system and user/group accounts. So we could bypass the credential management service in Multi-Cluster environment. For performing reliable and efficient sharing of computing resources between Classroom Clusters, we employ the meta-scheduler - GridWay [16] to provide a cooperating functionality.

GridWay is an open-source community project and it is highly modular, allowing adaptation to different infrastructures. We customized the prolog, wrapper, and epilog behavior in GridWay; moreover, we modified EM_MAD module to replace Globus [17] GRAM functions. Security is always the top priority for online services. We've replaced the GridFTP [18] with general SFTP/SCP service to migrate user data between head nodes. By using GridWay which makes central job submission and job dispatch to computerized classrooms in different geographical locations possible. Figure 5 shows our approach how to work with the Classroom Cluster services. Like the way using a typical PC cluster, users need to prepare simple job script and submit it to the Dispatch Manager via command line interface. The Dispatch Manager will parse the job script by its scheduling policy, and then will be transparently converted to equivalent Torque job script in head nodes. User's job files will be archived as a tar file and transferred to head node's working directory for execution when Dispatch Manager dispatched them.

Open Science Index, Computer and Information Engineering Vol:6, No:5, 2012 publications.waset.org/1719.pdf

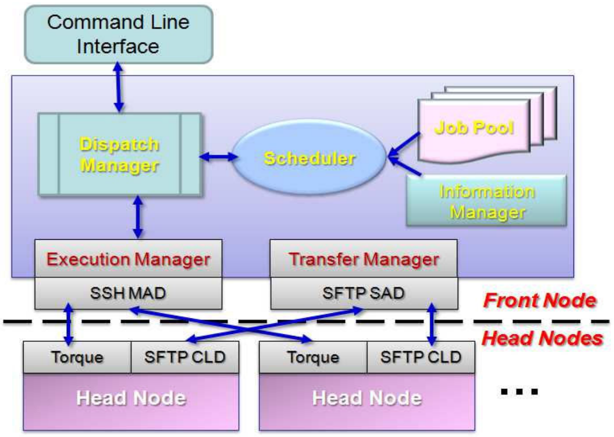

Fig. 5 Multi-Cluster collaborative flowchart

Hosting the meta-scheduler prevents us from installing additional software on remote head nodes. This also means that our system has the scalability for new PC classrooms to join it. So we could add new PC classrooms to raise the total computing capacity in the future.

V. Performance Evaluation and Analysis

To evaluate the performance of Classroom Cluster in a PC classroom, we employ HPL (High Performance Linpack) [19] and compare the results with two PC clusters in NCHC. The two PC clusters are: the self-made cluster named Siraya [20] and the Pilot Cluster. The details of hardware specification are listed in Table I.

TABLE I
Hardware Specification Of The Clusters

	Classroom Cluster	Pilot Cluster	Siraya Cluster
Processor	Intel Core2 Duo 2.83GHz x1	AMD Opteron 2356 Quad-Core 2.3GHz x2	AMD Opteron 275 Dual Core 2.2GHz x2
Memory	8G DDR2 SDRAM	8GB DDR2 ECC SDRAM	8GB DDR400 ECC SDRAM
Network	Gigabit Ethernet x1	Gigabit Ethernet x1	Gigabit Ethernet x2, 10Gb/s InfiniBand
Total Nodes	PCs x40	1U Rack-mount servers x12	1U Rack-mount servers x80

HPL is to solve a dense linear system problem and is the widely used benchmark for evaluating performance of supercomputer systems. However, the performance of PC cluster is largely application-dependent. We also use the NCHC Benchmark Suite which contains five benchmarks, namely hubksp, nonh3d, bem3d, ns3d and jcg3d, picked from four application domains. The hubksp program comes from physics, and nonh3d is an atmospheric science case. Both bem3d and ns3d are applications of parallel computing in the field of Computational Fluid Dynamics (CFD). The last one, jcg3d is from computational solid mechanics. These particular jobs helped us to distinguish the performance in different problem domains.

Table II shows the maximal performance (Rmax) and efficiency of three clusters on running 80 processors HPL job. We observe that the Pilot Cluster presents the best result among these machines, while Siraya is the worst one. Although this diskless based Classroom Cluster is not well-tuned on networking parameters, but it is capable of running parallel computations.

TABLE II
HPL RMAX And Efficiency

Cluster Name (Network)	Rmax in GFlops	Efficiency (%)
Classroom Cluster (GbE)	271.1	30
Pilot Cluster (GbE)	434.24	59
Siraya (IB)	271.05	77
Siraya (GbE)	228.8	65

From sequential elapsed-time results listed in Table III, the Pilot Cluster excels Classroom Cluster while running single core except the jcg3d and nonh3d project. The Classroom Cluster proved that it could provide capability for scientific computing. According to the floating point operations per clock cycle information, the Intel Core2 Duo doubles the AMD Opteron processor. For sequential jobs, Siraya apparently will be the worst one due to its less powerful processor. In terms of parallel processing, we evaluate hubksp and nonh3d jobs on many processors. Figure 6 and Figure 7 show the performance comparison of the Pilot, Siraya and Classroom Cluster. It is a comfort to see that Classroom Cluster took less time than both of the Pilot and Siraya cluster while executing nonh3d. For parallel hubksp jobs, traditional cluster systems including the Pilot and Siraya cluster edged out Classroom Cluster on 32 cores. There is a plausible reason for results of Classroom Cluster while running the hubksp case beyond 16 cores. The main cause for a drop in performance is NFS network contention with the computing channel over the same switch in the diskless environment. Both Siraya and Pilot Cluster have been separated sub-networks for monitoring, computing and file system to avoid conflicts. Current desktop PCs are not dedicated for high performance computing. However, for non urgent jobs, the Classroom Cluster will be qualified.

TABLE III
Performance Of NCHC Benchmark Suit

	bem3d	hubksp	jcg3d	nonh3d	ns3d
Classroom Cluster	46m27s	126m2s	44m24s	59m58s	162m14s
Siraya	94m50s	189m3s	102m1s	149ms3s	272m15s
Pilot Cluster	40m38s	43m6s	98m45s	64m35s	42m18s

Open Science Index, Computer and Information Engineering Vol:6, No:5, 2012 publications.waset.org/1719.pdf

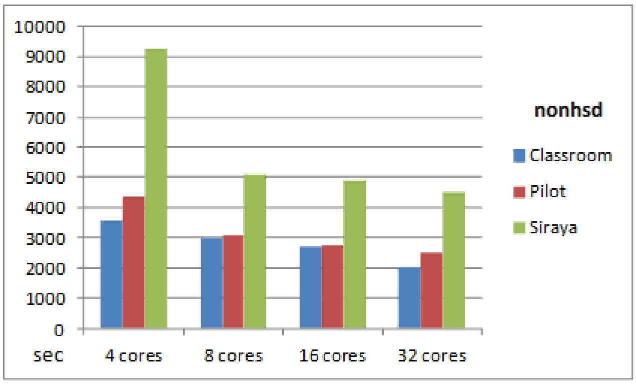

Fig. 6 NCHC Benchmark Suit - nonh3d

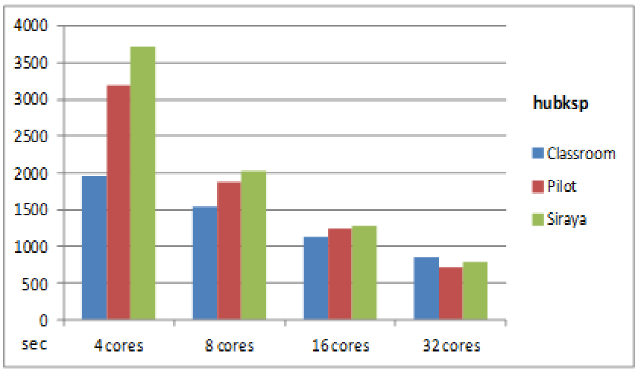

Fig. 7 NCHC Benchmark Suit – hubksp

VI. Conclusion

It is apparent that PC clusters are becoming one of the most important platforms for high-performance computing because of the advantage of delivering high-performance at low cost. In this paper, we present an innovative way and automated mechanism to deploy the pre-configured cluster computing environment that can utilize the existing computing power residing in PC classrooms. It works very well especially for sequential or parallel computationally bound applications. But it does not apply to communication or I/O bound applications which need high performance file system or local scratch space on computing nodes. The modified meta-scheduler is also introduced to perform reliable and efficient sharing of computing resource resides in geographically distributed PC classrooms. For general science applications, the benchmark results indicate that our development not only has the advantage of better cost/performance ratio but also has the potential to outperform high-end machines. In addition, the time and manpower required to build and manage a computing platform in geographically distributed PC classrooms also can be reduced by this development.

Acknowledgment

This work is supported by National Science Council, R.O.C., under the contract number of NSC-100-2221-E-492-015.

References

[1] R. Buyya (ed.), “High Performance Cluster Computing: Systems and Architectures,” Prentice Hall, 1999.
[2] C. Reschke, T. Sterling, D. Ridge, D. Savarse, D. Becker, P. Merkey, “A Design Study of Alternative Network Topologies for the Beowulf Parallel Workstation,” Proceedings of the Fifth IEEE International Symposium on High Performance Distributed Computing, (1996).
[3] T. Sterling, D. Becker, and D. Savarese,“Beowulf: A Parallel Workstation for Scientific Computation,” Proceedings of the Fourth 1995 International Conference on Parallel Processing (ICPP), Vol. 1, pp.111-14 (1995).
[4] A. Apon, R. Buyya, H. Jin, and J. Mache, “Cluster Computing in the Classroom: Topics, Guidelines, and Experiences”
[5] Message Passing Interface Forum, “MPI: A message-passing interface standard,” International Journal of Supercomputer Applications 8 (3/4) (1994) 165-414
[6] P. M. Papadopoulous, M. J. Katz, and G. Bruno, “NPACI Rocks: Tools and techniques for easily deploying manageable Linux clusters,” In IEEE Cluster 2001, October 2001.
[7] P. Uthayopas, T. Angsakul, and J. Maneesilp. “System management framework and tools for beowulf cluster,” In Proceedings of HPCAsia2000, Beijing, May 2000.
[8] Open Cluster Group, “OSCAR: A packaged cluster software stack for high performance computing,” http://www.openclustergroup.org.
[9] C. Y. Tu, W. C. Kuo, Y. T. Wang, Steven Shiau, “E2CC: Building energy efficient ClassCloud using DRBL,” in Grid '09: Proceedings of the 10th annual international conferenceon Grid Computing. Banff, AB, Canada: IEEE Computer Society, 2009, pp. 189-195.
[10] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, “Xen and the Art of Virtualization,” In Proc. 19th SOSP, Lake George, NY, Oct 2003.
[11] PXELINUX, http://syslinux.zytor.com/wiki/index.php/PXELINUX.
[12] NCHC, National Center for High-performance Computing, http://www.nchc.org.tw
[13] Wake-on-LAN, http://en.wikipedia.org/wiki/Wake-on-LAN
[14] TORQUE Resource Manager, http://www.clusterresources.com/products/torque-resource-manager.ph.
[15] I. Foster, C. Kesselman, S. Tuecke, “The Anatomy of the Grid: Enabling Scalable Virtual Organisations',” International Journal of High Performance Computing Applications, 15 (3). p. 200-222. 2001
[16] P. Armstrong, “Building a Scheduler Adapter for the GridWay Metascheduler,” Faculty of Engineering Summer 2006 Work Term Report
[17] I. Foster and C. Kesselman, “Globus: A Metacomputing Infrastructure Toolkit,” International Journal of Supercomputer Applications 11 2 (1997), pp. 115-128.
[18] The Globus Project: the GridFTP protocol, http://www.globus.org/datagrid/gridftp.html.
[19] A. Petitet, R. C. Whaley, J. J. Dongarra, and A. Cleary. “HPL - A Portable Implementation of the High-performance Linpack Benchmark for Distributed Memory Computers,” Available: http://www.netlib.org/benchmark/hpl/
[20] Siraya Cluster, Available: http://siraya.sro.nchc.org.tw/