FACTA UNIVERSITATIS (NIŠ)
SER.: ELEC. ENERG. vol. 18, No. 2, August 2005, 205-218

Using a Hybrid platform for Cluster, NoW and GRID computing

Dimitris Kehagias, Michael Grivas, and Grammati Pantziou

Abstract: Clusters, Networks of Workstations (NoW) and Grids offer a new, highly-available and cost-effective parallel computing paradigm. Their simplicity, versatility and unlimited power made them a rapidly adopted technology. It seems that they form the new way of computing. Although these platforms are based on the same principles, they differ significantly, needing special attention to their characteristics. Future computer scientists, programmers and analysts have to be well prepared, both about administrative and programming issues, in order to ensure faster and smoother transition.

We present the architecture of a dynamic clustering system consisting of Beowulf class clusters and NoW, as well as our experience from constructing and using such a system, as an educational infrastructure for HPC.

Keywords: Beowulf clusters, noW, dynamic cluster, GRID, computer science course.

1 Introduction

Since NASA researchers used PCs [1] to implement a high performance system until today, the High Performance Computing (HPC) community witnessed a diversion from expensive, proprietary parallel systems to complexes of common computing devices, to a gigantic distributed system built on top of the Internet. Such platforms have also enabled smaller organisations build or participate in Clusters or multinational Grids [2]. Clusters share conventional technology, they are easier to understand and administer and they offer unrivalled availability and are largely expandable ([3], [4],[5]).

Education did not follow. Since Clusters and NoW are made of common computers, there is a feeling that no new courses or material is required, as happened

Manuscript received Feb 27, 2005

The authors are with Department of Informatics, TEI of Athens, 12210 Aigaleo, Greece (e-mails: `dkehayas@teiath.gr`, `mgrivas@cs.teiath.gr`, and `gpantziou@teiath.gr`).

with transputers or vector computers. This thought lacks the consideration of idiosyncrasies, stemming from the complexity of such systems. The new paradigms of computing have to be presented to the students of higher education. On the other hand, many educational organisations lack the luxury of large budgets and cooperation with other, larger organisations may not always be feasible. After NASA's dedicated Cluster, others have seen existing computing facilities as possible Clusters, steeling idle cycles ([6]). PCs and workstations together -permanently or temporary, can realise a costless, powerful system. On such a system, students can work on distributed computing, HPC, parallel programming and Grid computing.

We propose a complex system made of conventional material and existing equipment, as a dynamic platform for research and education. Students have the opportunity to see, use, program and administer a complex platform. This paper is the sequel of [7] and [8], where a first approach to a mixed architecture and its educational benefits was made. Section 2 analyses the current form of the proposed architecture, describing also the technology behind it. Section 3 explains how the proposed architecture is embedded in the syllabus and is used in the courses. The outcome is presented in section 4, including novel issues that affect contemporary curricula (subsection 4.1) and its merits to the students' education and understanding. This paper concludes in 5.

2 The Proposed Architecture

In this section we will present the dynamic Cluster system built in our Department, an extension of the one described in [7]. The basic platform uses the equipment of a laboratory and a Beowulf class Cluster built specifically for this system. Then, more laboratories may participate, including independent workstations, whole NoW and other Clusters. There is a single job entry point, the console, that controls both the NoW and the Cluster. The console may also communicate with any additional system, either directly, i.e. messages to each workstation, or via another central entity, like it happens with other clusters. The proposed architecture provides solutions to the security problem for the NoW and remote connections, without compromising the performance in the local Clusters. Additionally, it demonstrates various methods of communication and involves complex administration tasks.

It must be noted that a single job entry point does not affect the functionality of the system as a tool for education. Whatever the task or the number of students, the submission may happen from the same single entry, without affecting the perception or the scope of a course. Each student is ultimately using always a single system as job entry or administration console. There are some works that seek the combination of clusters and NoW or other distributed systems, especially as parts of a Grid [9]. In most of those cases, the cluster participates to the whole as a single powerful computer [10]. In a small environment, where the number of workstations in a NoW is comparable to the number of the Cluster nodes, this model seems unsuitable. However, our proposed model embraces such structures. As seen in Fig. 1, some parts of the whole system may be independent clusters or NoW that are seen by their central controller, as a single system.

We should notice that although heterogeneous computers can coexist in a NoW or Cluster, in most of the cases it is preferable to use homogeneous computers [9], meaning absolutely compatible and with similar performance. Some rules for hardware selection have been already delineated in [11], [12], [13], [14] and [15]. The common PCs based on Intel or compatible CPU have the best price/performance ratio and the laboratory equipment is such. A dual or quadruple CPU would be preferable ([12]) and may be included later, either forming new Clusters or as a member of our existing platform or as the console.

The following subsections describe the constituting parts.

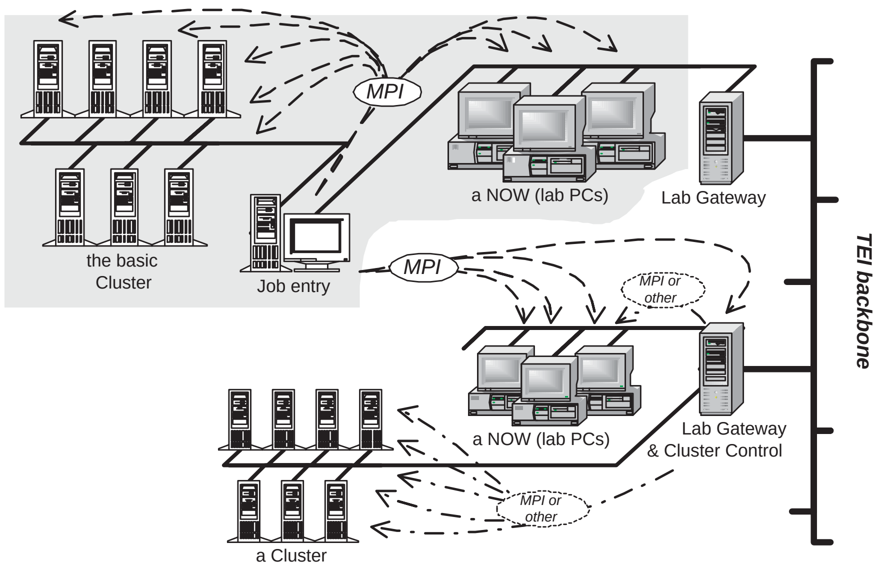

Figure 1. The architecture of a dynamic Clustering system.

2.1 Clusters

Since 1988 ([16]), it has been seen that, under specific circumstances, a bunch of independent computer interconnected over a fast network, can produce high performance, comparable to that of traditional supercomputers. The usual form of cluster is a power-of-two amount of homogeneous computers that are connected on a fast network, although they do not have to be homogeneous, neither on a fast network. Nodes may be workstations, PCs or special hardware designed for HPC. Communication solutions include common Ethernet (100 Mbps or 1Gbps), special HPC networks such as Myrinet, SCI, InfiBand etc, or even specially designed hardware. A very low budget cluster may involve 4 PCs and a 100 Mbps Ethernet, while NASA's future, gigantic Cluster involves 10240 Intel Xeon processors, 500 Terabyte storage and memory bandwidth of over 1 Terabyte per second (announced recently: `http://www.sgi.com/company_info/newsroom/press_releases /2004/july/supercomputing_ctr.html`).

The major advantages of clusters over traditional proprietary supercomputers are: The cost is significantly lower, they may offer higher availability, they require cheaper (common) administration and they may be more expandable. Single disadvantage is that they do not offer the same performance, due to high latencies in communication. This disadvantage can be rendered negligible with a proper compound of IO-bound and process-bound tasks ([17]). Then, contemporary microprocessors employ a load of other technological advancements for high performance ([18]). Among the most popular forms of Clusters, are the Beowulf class clusters [12]. They are dedicated, homogeneous clusters, that are deployed when performance is the main priority. The major two features of such clusters are common PCs as nodes and homogeneity, i.e., all of its computing nodes are identical dedicated nodes. Exception may be the ”front-end” computer, which allows users to submit jobs to the system. The rest of the nodes are usually not directly accessible by the users; they do not have keyboards, mice, or monitors.

Our principal Cluster consists of 7 nodes and the server. Each node is a Pentium 4 PC at 1,8 GHz with 512 MB RAM and a 32-bit PCI Ethernet NIC, connected to a 16-port switch. Static configuration to 100 Mbps, full duplex, ensures best performance and lessens undetected performance degradation caused by faulty parts. Each node has a 40 GB EIDE hard disk drive for boot-up and future DB and file purposes. They are configured with MPICH 2, MPICH 1.2.4, PVM and LAM, for future use. The Operating System (OS) is Linux, kernel 2.6.8, installed with minimal software, including the absolutely necessary communication and execution software, such as rsh, and tools for configuration. There is no swap partition for our nodes, since nodes are preferable to work within their memory limits to avoid hiding faults by erroneous programs, wrong configuration, system faults etc ([9],[19]).

In theory, any similar workstations within TEI can also be driven by our server, as members of the Cluster. For performance and latency reasons, this is never done. Any other Cluster may participate to this platform, over the TEI backbone (or even the Internet). They may be indirectly attached to the Console and be accessed using MPI messages. That includes autonomous workstations and integrated complexes like NoW or Clusters.

2.2 Networks of workstations (NoW)

NoW are complexes of computers -in the form of clusters- that are designed to take advantage of otherwise ”wasted” computing cycles on unused or lightly-used computers. A master system takes job requests and submits them for execution on workstations that have available resources to execute them. The key differences between a NoW and a Beowulf cluster are that workstations are not dedicated (i.e. autonomy in work), they do not obey to central control and they may exhibit heterogeneity (i.e. varying characteristics or architectures). However, Beowulf class Clusters and NoW are often considered as a single model of parallel systems, because they share common characteristics, such as:

- They both consist of common workstations or PC, connected via a common network, usually an Ethernet LAN or so.
- They have similar hindrances and limitations that include high latencies, distributed memory and possible unavailability of nodes.
- They are programmed in similar way, usually message passing and most of the times using the same tools (MPI, PVM etc).

Nevertheless, a NoW and a Cluster may be complementary platforms. Using the laboratory equipment, we have 18 PCs, all employing Pentium IV, 1.4GHz or better, and 512MB RAM. Those PCs are used for several tasks and courses, include Microcomputers and Applications, Visual Basic, other programming languages, Signal Processing and others. In order to cover all needs and requirements, they are configured with two operating systems. One is Linux with kernel 2.6.8, compatible to that of the Cluster, but a full installation including X-Windows, applications and tools for programming etc. When working with Linux, each PC can run assigned tasks from the server, in parallel to any local user tasks.

The laboratory communication infrastructure is an IP-based network over 100 Mbps Ethernet. Via a gateway, that laboratory network is connected to the TEI backbone, where it can contact Internet or other systems within TEI.

2.3 The server or console

As mentioned before, it is one of the Cluster nodes, employing monitor, keyboard and mouse. It runs Linux, configured as both a member of the Cluster and the NoW, with full installation and extra control tools. The server carries two Ethernet cards, eth0 serving the Cluster and eth1 connected to the laboratory network. This way, it can control both the Cluster and the NoW, ensuring isolation between the Cluster and the laboratory network. Any Cluster node is administered using rsh.

Among the extra software installed, there is a Web server and other tools for scheduling over Clusters, detection of systems performance over a network, network use, remote control facilitation and Cluster maintenance programs.

2.4 Technicalities

Local networking has been based on Fast Ethernet, which offers best price / performance. High-performance cards, such as Myrinet (www.myrinet.com) and Dolphin SCI (`http://www.dolphinics.no/`), offer incomparable advantages, like up to 8 Gbps bandwidth and minimal latency (6.3 μsec), but their cost is overly high. Gigabit Ethernet offers a 1 Gbps bandwidth, but results from independent studies (e.g. from the Scalable Computing Laboratory, Iowa State University, `http://www.scl.ameslab.gov/`) show no real gain. Other solutions have been proposed in [12], but most are in experimental stage.

The middleware for parallel computations (using message passing) chosen is the MPICH 2 (`http://www-unix.mcs.anl.gov/mpi/mpich`). The selection has been driven by the fact that there had been some experience with MPICH 1.2.4 in configuring and using it. A prime role played the fact that it is one of the most commonly used libraries. We have also installed other solutions, such PVM (`http://www.netlib.org/pvm3`) and LAM (`http://www.lam-mpi.org`), for future use.

Last but not least, the Operating System of choice is Linux. We tried several distributions ([7]), but finally this is not of great importance since it has to be rebuilt for maximum performance. We have also updated to newer kernel versions. Other candidates, like Sun Solaris, FreeBSD, MS-Windows and BeOS, were dropped either because of lack of support or because of lower performance and security.

3 Incorporation Into the Courses

The versatility of the proposed platform enables courses to offer more hands-on experience on diverse, complex architectures and demonstrate their functionality and different connection models. Until now, we have not incorporate the use of this platform to as many courses as possible or even desired. The most appropriate and related courses, though, have been Grid-enabled using it. In addition, non-typical modules or educational unities that may be covered or involved in other courses, have also embrace the merits of it. Following is the list of the most related subjects, that may be courses or not. The list does not exhaust the possible uses of the platform.

3.1 Parallel programming

Parallel programming course in our department covers a wide band of issues regarding parallel computations. It investigates the properties of parallel programming per se, its routes and theories, the infrastructures and the underlying mechanism that realise it. Regarding the programming issues, it presents the models of parallel computation and general principles and specific parallel algorithms. It is important that a significant part of the course is dedicated to scheduling, networking and communications and the MPI.

As in most of the courses, there are also workshop sessions, where students are required to produce code for specific tasks. Their programs use the MPI libraries and may be executed on a local PC or against the proposed platform. In addition, changing the number of participating computers and comparing their parallel program against the serial version, they can study the effect of parallelism, the bounds of performance advancement (Amdahl Law), the interference of other issues (latencies, workload etc) and in general, the students can acquire a very clear understanding of both parallel code and the details of parallel execution and its underlying materials.

3.2 Operating systems II

In this course, the students receive an in-depth knowledge of distributed systems. Having possessed the necessary basic knowledge from its preceding course (OS I), they have the opportunity to focus on more advanced issues and to emphasize in problems and properties that are beyond common single-processor machines.

The parts of the course that are related to the proposed platform include distributed systems hardware and software, networking, communication protocols, remote procedure call (RPC), message passing, task scheduling and synchronisation, atomic transactions, multi-threading, distributed file systems and security issues. It is important that the media to that is Unix and its variances, which is the basis of our platforms software. Unix offers a nice educational tool for demonstrating different architectures, system administration and security and networking. Our platform can demonstrate each of those points, both from a programmers view and from the administrators perspective that includes software and hardware knowledge.

The platform uses a complex of software that includes Linux, its configuration and its administration, RPC over unprotected and secure connections, networking issues in two levels, including insecure high-speed local connection and secure communication over a common LAN, in a highly insecure environment. A simple installation and configuration of this system gives a very clear understanding of many issues and technical details, in every aspect of distributed systems.

3.3 Distributed computing

This is not a course of itself. It participates, however, as a set of ideas, knowledge and information in many other courses, including parallel programming, operating systems, system security and introductory courses. The majority of its characteristics have already been proposed in the previous subsections, including tasks management and systems communications.

A more specific part involves test and performance measurements. For a detailed scrutiny of performance and functionality, there is a large set of tests and benchmarks that study specific characteristics of a distributed system and of parallel programs, like the ones propose in [12] and [14].

Other issues include security, which is a major area of investigation, especially as a part of Grid computing. The proposed platform offers a wide variety of problems and point of discussion, regarding security. It has not been yet embedded into the existing curriculum, but it is discussed later.

3.4 System administration

Again, this is not a real course, but a non-typical educational module, participating in different courses. System administration includes all the necessary knowledge and experience required to install, configure and maintain a system in well working condition. It involves knowledge in diverse areas of computing that span systems security, networking and communications, operating systems principles, programming principles and hardware.

The diversity of equipment that participates in the proposed platform, offers an excellent demonstration and experience acquisition tool. While installation and configuration in networking equipment or a server may be a burden and difficult, it is a minimum requirement for setting up such a system. In all the phases of that systems platform, students will see every single part of installation and configuration regarding the aforementioned areas. Its maintenance, too, offers an excellent occasion for study of complex systems, its hardware, its operating system and its application software.

3.5 Final projects

In the final year, students must conduct their own research in fields relative to their studies and expertise. They are assigned specific projects by professors that follow and control their progress. Those projects are usually in the form of application software or more scientific investigation.

Any project related to Distributed Computing and Databases, Operating Systems, Systems Administration and especially Parallel Programming, can be helped broadly by the use of the proposed platform. Instead of simulation results and solely bibliographical research, students have the opportunity to try out their works. Furthermore, when projects regard Grid computing, in any of its forms, the availability of a tool to try, under real conditions, boosts the research and the understanding that students acquire.

4 Outcome: Merits, Shortcomings and Probable Disadvantages

Some of the most interesting characteristics of the proposed platform are:

- Physical and logical separation of the primary Cluster and the NoW and other systems.
- Independent work in each platform.
- The set of jobs is shared among the nodes of the Cluster and the NoW as in a single system.
- Administration is independent.
- There is no need for a single, common protocol.

This combination of characteristics offers a wide variety of configurations and methods of work. MPI and the rest of the software we use, extends even further the flexibility and versatility of the platforms functionality. Several logical interconnection, e.g grid, mesh or torus, a variety of scheduling methods, different synchronisation models and other architectural or functional attributes can be studied in a real working environment.

4.1 New paradigms requirements in contemporary education

Computing Grids have been out and working for quite some time, although education has not reacted accordingly. The basics of it may be covered by most curricula that involve parallel programming as a major course. However, there are certain details of Grids that need special attention and care. In a common, basic course of parallel programming, pupils are not asked to investigate alternatives in scheduling or the problems that may impose one or another form of distributed computing.

It has been proposed that simulation may serve better for education ([20],[21]), since it is controllable and can demonstrate in details the process of each task. Although simulation is a very powerful tool, it cannot exhibit all factors that affect a distributed system's performance, neither does it provide any help to administration of complex systems. Simulators provide a "sterilised" environment, where any interference or affecting factor is under strict control. There is always a possibility that in real environments things may go differently of the expected, either because of bad calculations or due to factors that have not been -or could not have been- considered before. Furthermore, technological education has always to provide the means to experience on administration tasks, such as installation, configuration, management and problem detection and solving. Such issues are exactly what simulation helps to avoid.

Thus, a system that can be as close to a real distributed systems demonstrating most -if not all- of a systems idiosyncrasies, while been under total control would be the most helpful for both parallel programming and administration on distributed systems. Such a system should be able to emulate the functionality of simple Clusters, NoW and more complex architectures, resembling to Grid as much as possible.

Several issues have been studied thoroughly before, like distributed memory, but seldom presented in education as concerns for programmers. Other characteristics have never been introduced as possible problems. Things become more complex in Grids, which introduce time differences and huge latencies. Programs that are monolithic, without deep consideration of parallel and distributed work are doomed to take no or little advantage of new technologies. Simple techniques, such as threads, cannot help alone. The major new challenges include:

- Distributed ownership: Resources are governed by several proprietors, with varying schedules etc.
- Heterogeneity: Resources may be of any kind, even outside the computing world, that need new techniques and methods to reveal their potentials.
- Dynamic resource group, in type, number and formation: Each constituting organisation acts autonomously; resources or communication may change any time.
- Absence of fixed resources patterns or schema: Resources have completely disparate attributes and behaviour. Predictions based on known patterns are not usually applicable.
- Enormous latencies, not controllable, not predictable: Communication may happen over the Internet.
- Unavailability: Resources and communication channels may become unavailable any time, before or during a process.
- Stale information: Information about resource load and functionality may not exist or get lost or come late.

- New costs: The use of a communication channel or a resource may cost in the broad sense. It may even cost money in the narrow sense.
- Security, for both resources and jobs. Neither hosts, nor code or communication channels may be considered trusted.
- New form of programming, new layers: The structural layers of a Grid may differ drastically from conventional programming ([4]).

Such challenges imposed that common techniques and traditional distributed computing methods are not appropriate to enable Grid computing. Although, there are a handful of new problems and issues that must be addressed in order to work with Grids, the potentials of this new paradigm are such, that it has been pervasive to practically all aspects of computing, including data storage, access and manipulation (Data Grids [22], [23]), distributed, multi-source information retrieval (Grid Information services [24]) and of course processing (Computation Grids [25], [26]), either in science or even the market ([27], [28]).

4.2 Further involvement

Next step is to evaluate the outcome of its use in the aforementioned courses and cognitive units. A principal consideration is to embed the use of this platform into more courses, so as to offer to the broadest possible extent, in-depth experience and knowledge over distributed computers. More courses are following, but the ones that will consider its use in the near future include security and the introductory course of first semester.

Security of operating systems As mentioned in the Distributed Computing subsection 3.3, security is a major concern, especially as a part of Grid computing. The proposed platform offers a wide variety of problems and point of discussion, regarding security. They include secure connections and encoded data over non-secure networks, prevention of the isolation of the Cluster against outside attacks, system vulnerabilities, content protection against unauthorised access and others.

The primary merit from our platform is that the students have the opportunity to see, examine and study in details and in real environments, all those issues. They may see how to build and to ensure a secure system. They can even emulate attacks and problems that otherwise would be only theoretically approached. Finally, they can set, control and test varying protection schemes.

Introduction to Informatics In an introductory course for information technology students, the requirements and expectations are quite high. Students have to start with computer usage, but the basics of IT include System Architectures, Communications, Programming, Networking and Network protocols, Scheduling and Synchronisation, even in a simplified, primary level. Having seen the basics in these areas, the proposed platform offers a nice introductory system that can demonstrate all varying scopes and dimensions of Computer Science and Information Technology.

5 Conclusion

The proposed architecture that combines NoW and Clusters can support HPC in a dynamic way, where any available computer can participate as an additional node. Its versatility and flexibility in configuration and functionality, together with its dynamic, strong computational power, renders it a very helpful demonstration tool for educational purposes.

This platform has already become a part of the curricula of several courses, seminars and intermediate sets of education sessions. The results of its use will soon be evaluated and it will be introduced to more courses, wherever it is applicable.

References

[1] J. A. Kaplan and M. L. Nelson, "A comparison of queueing, cluster and distributed computing systems," NASA Langley Research Center, Tech. Rep. TM 109025 (Revision 1), June 1994.

[2] I. Foster and C. Kesselman, *The Grid: Blueprint for a New Computing Infrastructure*. Morgan-Kaufman, 1999, ch. ch. 2. Computational Grids.

[3] C. Ernemann, V. Hamscher, U. Schwiegelshohn, A. Streit, and R. Yahyapour, "On advantages of grid computing for parallel job scheduling," in *Proc. of 2nd IEEE Int. Symp. on Cluster Computing and the Grid CC-GRID*, 2002.

[4] M. Baker, R. Buyya, and D. Laforenza, "Grids and grid technologies for wide-area distributed computing," *Int. Journal of Software: Practice and Experience (SPE)*, vol. 32, pp. 1437–1466, Dec. 2002.

[5] I. Baird, "Understanding GRID computing," *Daily news and information for the global grid community*, vol. 1, July 202.

[6] A. Chavez, A. Moukas, and P. Maes, "Challenger: A multi-agent system for distributed resource allocation," in *Proc. of the First Int. Conference on Autonomous Agents (Agents'97)*, W. L. Johnson and B. Hayes-Roth, Eds. New York: ACM Press, 1997, pp. 323–331.

[7] D. Kehagias, M. Grivas, G. Meletiou, G. Pantziou, B. Sakellariou, D. Sterpis, and D. Ximerakis, "Building a low-cost high-performance dynamic clustering system,"

in *Proc. of the 6th IEEE Int. Conf. on Telecommunications in Modern Satelite, Cable and Broadcasting Services (TELSIKS 2003)*, B. D. Milovanović, Ed., vol. 2. Nis, Serbia: Faculty of Electronic Engineering, 2003, pp. 608–613.

[8] D. Kehagias, M. Grivas, G. Meletiou, G. Pantziou, B. Sakellarios, D. Sterpis, and D. Ximerakis, “A low-cost dynamic clustering system for education and research,” in *Proc. of TEMPUS Workshop*, 2003.

[9] H. Hellwagner, W. Karl, and M. Leberecht, “Enabling a PC cluster for high-performance computing,” in *Proc. of the 21st Workshop on Vector and Parallel Computing*, 1997, pp. 18–23.

[10] A. Reinefeld and V. Lindenstruth, “How to build a high-performance compute cluster for the grid,” in *Proc. of Int. Workshop on Metacomputing Systems and Applications (MSA’2001)*, 2001.

[11] “Improving load balancing property during system reconfiguration,” *New Advances in Computer Aided Design and Computer Graphics*, vol. 1 and 2, pp. 582–587, 1993.

[12] R. S. Morrison, “Cluster computing architectures, operating systems, parallel processing and programming languages,” Apr. 2, 2003.

[13] S. Paas, M. Dormanns, T. Bemmerl, K. Scholtyssik, and S. Lankes, “Computing on a cluster of pcs: Project overview and early experiences,” Nov. 1997.

[14] J. Lindheim, “Building a beowulf system.”

[15] T. Sterling, J. Salmon, D. J. Becker, and D. F. Savarese, “How to build a beowulf: A guide to the implementation and application of PC clusters,” May 28, 1999.

[16] M. Stumm, “The design and implementation of a decentralized scheduling facility for a workstation cluster,” in *Proc. of the 2nd IEEE Conference on Computer Stations*, 1988, pp. 12–22.

[17] J. Subhlok, D. R. O’Hallaron, T. Gross, P. A. Dinda, and J. A. Webb, “Communication and memory requirements as the basis for mapping task and data parallel programs,” in *Supercomputing*, 1994, pp. 330–339.

[18] T. Y. Liang, C. K. Shieh, and D. C. Liu, “Dynamic task scheduling on multithreaded distributed shared memory systems,” in *Proc. of the 1998 Int. Conference on Parallel and Distributed Processing Technique and Applications*, Las Vegas, USA, July 1998, pp. 1058–1065.

[19] R. G. Brown, “Maximizing beowulf performance,” in *Proc. of the 4th Annual Linux Showcase and Conference*, Atlanta, Georgia, USA, Oct. 10–14, 2000.

[20] M. Murshed and R. Buyya, “Using GridSim toolkit for enabling grid computing education,” in *Proc of the Int. Conference on Communication Networks and Distributed Systems Modeling and Simulation (CNDS 2002*, San Antonio, Texas, USA, Jan. 27–31, 2002.

[21] H. Casanova, A. Legrand, and L. Marchal, “Scheduling distributed applications: The SimGrid simulation framework,” 2003.

[22] W. Allcock, A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, and S. Tuecke, “The data grid: Towards an architecture for the distributed management and analysis

of large scientific datasets," *Journal of Network and Computer Applications*, vol. 23, pp. 187–200, 2001.

[23] G. Cancio, S. M. Fisher, T. Folkes, F. Giacomini, W. Hoschek, D. Kelsey, and B. L. Tierney, "The datagrid architecture version 2," 2001.

[24] K. Czajkowski, S. Fitzgerald, I. Foster, and C. Kesselman, "Grid information services for distributed resource sharing," in *Proc. of the Tenth IEEE Symp. On High Performance Distributed Computing (HPDC-10)*. IEEE Press, Aug. 2001.

[25] F. Giacomini, F. Prelz, M. Sgaravatto, I. Terekhov, G. Garzoglio, and T. Tannenbaum, "Planning on the grid: A status report," Oct. 18, 2002.

[26] J. Basney, M. Livny, and P. Mazzanti, "Harnessing the capacity of computational grids for high energy physics," in *Conf. On Computing in High Energy and Nuclear Physics*, 2000.

[27] I. Foster, C. Kesselman, and S. Tuecke, "The anatomy of the grid: Enabling scalable virtual organizations," *Int. J. Supercomputer Applications*, vol. 15, no. 3, 2001.

[28] H. Casanova, "Distributed computing research issues in grid computing," *ACM SIGAct News*, vol. 33, no. 3, pp. 50–70, 2002.