A Distance Learning Tool for Teaching Parallel Computing[1]

RAFAEL TIMÓTEO DE SOUSA JR., ALEXANDRE DE ARAÚJO MARTINS,
GUSTAVO LUCHINE ISHIHARA, RICARDO STACIARINI PUTTINI, ROBSON DE OLIVEIRA ALBUQUERQUE
Electrical Engineering Department, Network Engineering Laboratory
University of Brasília
Caixa Postal 4386, 70919-970 Brasília – DF, Brazil
BRAZIL
desousa@unb.br, amartins@vma.com.br, luchine@hotmail.com, puttini@unb.br, robson@redes.unb.br
http://www.redes.unb.br

Abstract: - This paper describes a www application developed to access and manage a virtual laboratory for teaching parallel computing. The laboratory is a cluster of personal computers with Intel processors running the NASA's Beowulf cluster software and using the MPI application-programming interface under the Linux operating system. This paper describes the construction, configuration and optimization of the Beowulf cluster as well as some theoretical concepts and real experiments implemented during the project. The motivation for using the Beowulf cluster under Linux comes from its excellent cost-performance ratio, which is an important factor in a teaching environment. The virtual laboratory environment consists of an Apache web server installed with PHP and MySQL modules, which were used for the application development. This environment can be accessed from Internet web browsers allowing users to remotely submit and run parallel programs on the implemented cluster.

Key-Words: - distance education, parallel computing, cluster.

1 Introduction

A parallel processing cluster is a group of independent computers interconnected by a network and working together in a very large problem that was partitioned in smaller ones, this way allowing the solution of problems formerly considered too big for a single computer [1]. Such processing configurations are very interesting for educational organizations that deal with research and teaching activities in several knowledge areas.

Why to build and configure a computer cluster if there are already supercomputers available in the market? The answer is very simple: the costs. There are not so many institutions that can afford to buy and maintain a supercomputer. The idea of developing parallel systems using commodity computers was first explored by Donald Becker and his colleagues at NASA [2]. The first Beowulf cluster was built with 16 486DX-4 nodes at the NASA's Goddard Space Flight Center [2]. The success of the project allowed other institutions to have great processing power at low cost, making this kind of cluster very popular.

Beowulf clusters have been designed and used by programmers with little or no experience in parallel programming. In fact, Beowulf clusters allow many universities, which usually have limited resources, to implement an excellent platform for teaching and research activities on parallel programming, providing teachers with high processing power at effective low cost. There are four main reasons that explain this minimal cost: (i) most interested students already have the GNU/Linux installed in their computers; (ii) clusters are built using commodity hardware or even hardware formerly considered obsolete; (iii) laboratory implementation and parallel programming studies are included in the process of students formation; and (iv) the utilization of standard libraries for parallel programming such as MPI (Message Passing Interface) [3] and PVM (Parallel Virtual Machine) [4] allow parallel programs written and tested in a GNU/Linux cluster to run on commercial supercomputers (e.g., Cray or IBM) with little or no modification.

Two other key points are the maturity of the operational system used in Beowulf clusters. Given the GNU/Linux robustness and the standardization of parallel programming libraries. Programmers have the warranty that their programs will run in future Beowulf clusters independent of processor and network equipment manufacturer.

The Beowulf cluster architecture is based on the existence of a master node that controls and manages the processing of the slave nodes. There can be any desired number of slave nodes, given that the communications hardware limits are respected. The master node generates all the processes and passes them to its slave nodes to be executed. The master

[1] This work is sponsored by CNPq, the Brazilian government agency for scientific and technological development.

node must be capable of keeping all the slave nodes occupying their processors during all the time. The cluster network is usually isolated from other networks to eliminate any influence from external network traffic. Only the master node has a connection to an external network to allow users to submit and control the execution of parallel programs and to perform cluster management tasks.

The idea of a www based virtual laboratory extends the concept of a cluster with a new user interface, which makes the high computational power available to remotely located group and individual users. This way, the cluster can be accessed using any web browser, allowing the user to be at any place, physically distant from the cluster. Specially, this ability can be used in distance education, for example in the context of a parallel-programming course with the student submitting his code and getting his results on-line. Another example is the use of the cluster to reduce the time for rendering a three-dimensional image from its form and texture file. This operation, which usually demands hours of processing power in typical workstations, can be done in a few minutes in a cluster. In this case, teachers and students from the fields of design, architecture, or computer graphics, among others, could have significant benefits.

Actually, a cluster offers almost unlimited possibilities because of the potential applications it has in most fields of science, mainly where a high computational power is involved to accomplish a scientific task or to teach/learn a certain discipline.

2 Building a Beowulf Cluster for the Virtual Laboratory

Some personal computers (nodes) are necessary to build the virtual laboratory cluster, a configuration with four nodes being the reference cluster for this paper. Each node needs a network card such as ethernet or any other technology, according to the intercommunication hardware (hub or switch).

The minimum node configuration consists of a 486 processor with 16 Mbytes RAM memory and a 1 Gbyte hard disk. Computers based on 386 processors can also be used, but the performance penalty hardly justifies the configuring effort and the constant maintenance required.

Typical parallel applications ask for 32 Mbytes of RAM or more per node. External devices, such as the keyboard and the monitor, are needed for the configuration phase, but after that only the master node needs these devices.

2.1 Physical Installation

A local area network hub or switch is needed to interconnect the cluster's nodes, according to the topology shown in Fig. 1.

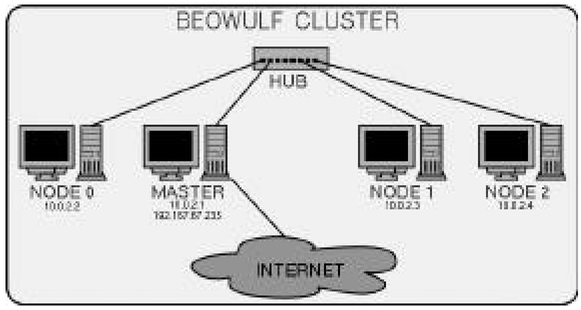

Fig. 1. The Virtual Laboratory Cluster Reference Topology

2.1 Software Installation

In each node the installation of the operational system GNU/Linux Red Hat 6.2 is organized in order to maintain the maximum of the system's resources available to the cluster. To fulfill this approach, only a minimum set of packages is installed in the nodes and all the other unnecessary services are disabled [5][6].

After the GNU-Linux Red Hat 6.2 installation, the Beowulf RPM packages are installed with a specific configuration for the master and the slave nodes. For the master node, a CD containing all the needed packages has been assembled allowing installation in a two-step process. First, using the step 1 CD directory the RPM program is updated, so that the other packages can be updated from the step 2 CD directory. Another similar CD was made for the slave nodes as well. All the necessary programs are freely available on the Internet and can be easily obtained. To boot the Beowulf kernel, a specific configuration file for the operating system initialization has been produced. Also, the MPI-X-PovRay computer graphics application for image rendering is installed, intended to serve as a basic parallel application example.

To complete the software installation, the Apache web server [7] is installed with the PHP and MySQL modules [8] enabled, allowing execution of the virtual laboratory environment application, which was developed as a new web based user interface for the cluster.

3 Testing and Optimizing the Virtual Laboratory Beowulf Cluster

Several cluster configurations were implemented. Especially the reference cluster configuration was tested to check out its correct functioning and to optimize cluster performance according to some known guidelines [9] [10]. Some operational results could be observed running the MPI-X-PovRay application [11] as described hereafter.

In particular, two clusters were built for testing, one with four identical computers and the other with four completely different computers.

In the homogeneous cluster, the general performance optimization was obtained turning off all the unnecessary services in all nodes, master and slaves. Using Beo-Status, the Beowulf management application [12], it was possible to see all the processors totally occupied at the running time, because the network technology was fast enough to transmit the MPI messages necessary to the completion of the work. The referred network has the 10Mbps cluster nodes interconnected on a 10Mbps ethernet emulated LAN in a 155Mbps ATM switch. This cluster concluded the task four times faster than a single computer.

In the heterogeneous cluster, which is representative of most of the real cases in institutions and universities, turning off the unnecessary services was not enough to get the shortest time in the completion of the task. So, attempting to achieve this goal, different computers (Table 1) were tested in the master node position, producing the performance results shown in Table 2.

Table 1. Computers Tested in the Master Node Position (Heterogeneous Cluster)

Computer	Clock (MHz)	Memory (Mbytes)	Hard Disk (Gbytes)
Pentium Thinkpad	150	48	2
Pentium 233	233	98	7
Pentium III	550	256	20

Table 2. Heterogeneous Cluster Performance Data

Master Node	Task Completion Time (s)
Pentium Thinkpad	26
Pentium 233	9
Pentium III	17

As can be seen in Table 2, with the heterogeneous cluster the best performance was achieved when the master computer was the Pentium 233 MHz. This is due to the fact that the master node only coordinates cluster activities and does not directly participate in parallel computing. So, choosing the Pentium III as the master node caused performance degradation, because the most powerful computer was no longer processing the tasks. Choosing the Pentium Thinkpad as the master node also degraded performance because this computer was unable to efficiently coordinate the slave nodes responding requests at a reasonable time. Although the best solution was achieved with the Pentium 233 Mhz, this computer was not able to distribute tasks so as to use the maximum capacity of the slave nodes (Fig. 2), but it obtained the best traffic management without markedly reducing the overall parallel computing power.

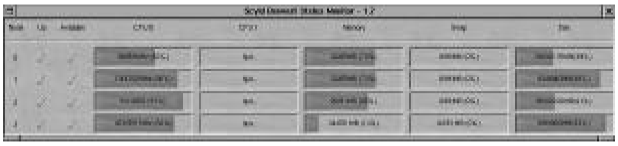

Fig. 2. Utilization of Node Resources (Measures from the BeoStatus Program)

Although a general approach for the configuration of heterogeneous clusters can not be drawn from these results, they show that cluster testing and optimization are essential for a better use of its computational power. For the same cost, the cluster performance changed dramatically with the choices made in its configuration. And because the virtual laboratory is intended to be used for a long period, any increase in performance without additional costs justifies the tests made as part of the designing.

4 A Web Based User Interface for the Beowulf Cluster

To make cluster services available on the Internet, a www application was developed in PHP language, with the objective of creating a complete environment to remotely manage and use the cluster with a very friendly user interface. The application was also produced as an effort to obtain a simple and effective distance education tool. The developed code could not be shown in this document due to space limitations, but some significant navigation screens are presented bellow.

Fig. 3 shows an Internet browser loaded with the project main page. Although this page is shown in Portuguese, the application can deal with pages translated into any language, because all the text is loaded with calls to variables, which can be changed in order to match the desired language text located in a specific directory. New versions are expected to support English and Spanish at least. Using the main page, a user, e.g. a student, may require a user's account or, if he/she already has one, the user can login by entering his e-mail and password. After the user authentication, the application opens the developed web interface for the cluster (Fig. 4).

The student can then run already existing parallel programs. Alternatively, the student can develop his own parallel programs at home (or in one of the university's programming laboratories) and submit them to be executed by the cluster. The web environment gives to student back all the results and files generated by their programs. Because parallel programming can be used to deal with a great variety of problems, the entire web environment can be used for teaching in several fields of science. A typical course could start with a parallel programming introduction and would go on to deal with a problem of a specific field. Another interest of this approach comes from the fact that a program tested in the cluster can be executed in another MPI environment, such as a supercomputer. This brings more quality to the teaching process, because the student can have access to a similar tool, but without incurring the same access restrictions and almost prohibitive cost.

For instance, the developed web interface could be helpful in a course on super-computer algorithms or parallel computing, or yet could be used for complex visualization processes and graphic representations (Fig. 5). This way, the virtual laboratory would allow teachers and students of parallel computing to illustrate parallel algorithms in a uniform manner, to teach/learn the development of portable parallel software, to use parallel libraries and to develop portable parallel software.

Fig. 3. Virtual Laboratory Main Page

Fig. 4. Cluster Access and Management Page (After User Authentication)

Thus, the developed application demonstrates an effective solution for a virtual laboratory that can be used for teaching parallel computing, which includes the remote submission and execution of parallel programs, given that the cluster is associated to a web environment such as the one implemented on this project.

Fig. 5. Presentation of Results from a Parallel Program Executed in the Cluster

5 Conclusion

With the implementation of a single Beowulf cluster, several teaching institutions and universities can provide courses in parallel programming or use parallel applications to solve theoretical or practical problems.

A virtual laboratory allows any teaching institution with only one computer connected to the Internet to use the high processing power offered by a cluster.

High performance commercial systems are very expensive, including not only purchase costs, but also support and maintenance costs. A Beowulf cluster has a low cost hardware maintenance and this cost can be shared by a group of institutions that use the cluster.

There are yet very few tools that can be freely used in the area of parallel computing and the available tools lack good technical support. There is a demand for qualified people that are able to create these tools and to provide efficient support. A virtual laboratory for teaching parallel computing can be a helpful tool and provides an adequate cost model to fulfill this demand.

The virtual laboratory also stimulates teaching and research activities in the field of parallel processing, providing a cheaper solution to access an expensive technological tool. As such, the implemented virtual laboratory is now being used in the network engineering laboratory at the University of Brasília - Brazil.

References:

[1] Spector, D.H.M.: Building Linux Clusters. O'Reilly, Sebastopol-CA (2000)

[2] http://www.beowulf.org. Beowulf Project Internet site accessed April 30, 2002

[3] http://www.unix.mcs.anl.gov/mpi/. Message Passing Interface Standard Internet site accessed April 30, 2002

[4] http://www.csm.oml.gov/pvm/pvm_home.html. Parallel Virtual Machine Internet site accessed April 30, 2002

[5] http://www.acm.org/crossroads/xrds6-1/parallel.html. Parallel Computing with Linux Internet site accessed April 30, 2002

[6] http://www.cacr.caltech.edu/beowulf/tutorial/building.html. Building a Beowulf System Internet site accessed April 30, 2002

[7] http://www.apache.org. Apache Software Foundation Internet site accessed April 30, 2002

[8] http://www.php.net. PHP Hypertext Preprocessor Internet site accessed April 30, 2002

[9] Evan, M., Hal, S.: Blueprints for High Availability: Designing Resilient Distributed Systems. John Wiley & Sons (2000)

[10] Buyya, R.: High Performance Cluster Computing, Vol.1. Prentice-Hall (1999)

[11] http://www.povray.org/. Persistence of Vision Ray Tracer Internet site accessed April 30, 2002

[12] http://www.scyld.com/. Scyld Beowulf Cluster Operating System Internet site accessed April 30, 2002