E-Learning Cluster Computers: a Self-Learning Approach with E-Contents

Hélio Vilas

Departamento de Informática, Universidade do Minho
4710 – 057 Braga, Portugal
Cei6482@di.uminho.pt

Abstract. This communication stresses some relevant features on interactive Internet services on distance learning, and presents an e-learning course proposal on "Cluster Computers" as a case study, with an in-depth analysis of its contents.

1 Introduction

Learning stands for a cognitive and/or physical process in which a person assimilates information and temporarily or permanently acquires or improves skills, knowledge, behaviours, and/or attitudes.

The development of new technologies, in special computers and the Internet, brought forth new demands and needs from potential consumers of knowledge, i.e., new 'clients' for learning. And so the concept of e-learning (learning by electronic means) has evolved.

E-learning covers a wide set of applications and processes, such as Web-based learning, computer-based learning, virtual classrooms and digital collaboration. It includes the delivery of content via Internet, intranet/extranet (LAN/WAN), audio and videotape, satellite broadcast, interactive TV and other electronic media.

The introduction of e-learning added new meaning to training and the possibilities for delivering knowledge and information to learners at an accelerated pace and opened a new world for knowledge transfers.

This communication aim to show what e-learning is about and features a proposal case study on Cluster Computers.

Clusters were designed to solve large computing problems in a much cheaper way than supercomputers. They are usually formed by a composition of hardware (nodes) and software components. Cluster nodes can be PCs, workstations, and SMP's.

Currently, clusters support applications raging from supercomputing and mission-critical software, through Web server and e-commerce, to high-performance database applications.

The nature of cluster computing allows illustrating the steps on an e-learning course design and how it works.

2 Why E-Learning?

Society has undergone such dramatic and fast changes over the last few years. Individuals feel compelled to keep up with its demands for well-informed, up-to-date and competitive citizens.

E-learning has many advantages over traditional learning:

- Reduction and rationalization of resources - one instructor (team) produces many materials that can be accessed anywhere, anytime, by many students;
- Temporal flexibility – there is the possibility of synchronous and asynchronous classes;
- Geographical independency - the students do not need to travel to school and there are no travel costs involved;
- Accommodative design to a variety of different interests and needs.

While the benefits of e-learning outweigh its possible constraints, attention must be paid to:

- Greater initial investments than traditional learning;
- Lack of physical personal teacher-student interaction;
- Some resistance to new methods and changes;
- Technological requirements, such as the client-server computer facilities, the Internet connections, and adequate supply of e-learning s/w environment.

The latter point is very significant since some technical issues may discourage e-learners; the two most frequent are low bandwidth and incorrect configurations.

As said above, online students must have certain technology available to them to be able to successfully work in an online environment. As an illustration, the lowest technical requirements are: a standard Pc with Internet Access; audio capability, and microphone.

When questioned, students tend to point flexibility as the greatest e-learning asset. They tend to choose online teaching over in person teaching because they can do it at their own pace and have always the ability to go back where they might not have understood and do it over [1].

At the moment big companies entirely understood the magnitude of e-learning and all inherent benefits, when they provide online continuing education for their employees.

3 Instructor-group Interaction

Although the concept of e-learning was developed to allow for students to reach for the materials by their own and anywhere away from school (and even knowing they are typically adult), special care must be taken when dealing with the interaction with the instructor. As good as the online materials may be, they are substitute for the human presence (human side) that the instructor represents. Moreover, the instructor offers additional flexibility that cannot be achieved by asynchronous materials.

Whenever is needed, during an online lesson; any student can ask questions sending an instant message. Even after that there are prearranged times when the instructor is available online.

Subsequent to the first online lesson, after knowing the instructor and all the group elements, feeling comfortable with the technological aspects and procedures everyone consider that there is much average support and empathy in the teacher-student interaction than it was alive.

4 E-Learning Evaluation

Like traditional learning, distance learning must contain feedback reports too. Must be planned at the beginning of course; the number of tests and exams to perform. All along the course, the group complete several self-tests, homework, quizzes and tests.

At the end at course students work on an exam or presents a final project that confer them the certificate award (guarantee that s/he got the knowledge or professional certification).

The course final exam may be online, on a pre-established location or at the instructing institution.

Another model of learning (in use in higher education and corporate training) combines e-learning with classroom or lab sessions; these can be particularly productive, assuming the learning model for each part has been carefully thought through.

5 E-Learning contents

E-learning materials and contents are designed by a multidisciplinary team aiming to ensure that the used methods and techniques properly support the course objectives. These resources are classified in two groups: synchronous and asynchronous.

The *synchronous* materials are any those accessed in real-time, comprising scheduled classes with the instructor, chat rooms dealing both with work in groups and clearing doubts and questions, instant messages to other fellow students and group members or to the instructor and online surveys and exams – a lot of online learning events in which all participants are logged on at the same time and communicate directly with the instructor and each other.

The *asynchronous* ones (tutorials) are those that can be accessed when the student wants to, so this interaction between teacher and students occur intermittently with a time delay. It contains text content pages online, audio and video lessons, group forums where students put their work and ideas before classmates for comment, self-tests, quizzes, homework and past exams.

6 Cluster Computers: Introduction to the Course Contents

Very often applications need more computing power than a sequential computer can provide. One-way of overcoming this limitation is to improve the operating speed of processors and other components so that they can offer the power required by those applications [2]. Even though this is currently possible to a certain extent, future improvements are constrained by the speed of light, thermodynamic laws, and the high financial costs for processor fabrication. A feasible and cost-effective alternative solution is to connect multiple processors together and coordinate their computational efforts. The resulting systems are popularly known as parallel computers, and they allow the sharing of a computational task among multiple processors.

There are three ways to improve performance [3]:

- Work harder,
- Work smarter, and
- Get help.

In terms of computing technologies, the analogy to this mantra is that working harder is like using faster hardware (high performance processors or peripheral devices). Working smarter concerns doing things more efficiently and this revolves around the algorithms and techniques used to solve computational tasks. Finally, getting help refers to using multiple computers to solve a particular task.

A cluster is a type of parallel or distributed processing system that consists of a collection of interconnected stand-alone computers working together as a single, integrated computing resource [4].

Clusters, built using commodity-off-the-shelf (COTS) hardware components and free, or commonly used, software, are playing a major role in solving large-scale science, engineering, and commercial applications.

One well-known example of distributed or clustered computing is the ongoing SETI (Search for Extraterrestrial Intelligence) project, in which thousands of users are sharing their unused processor cycles to help search for signs of "rational" signals from outer space.

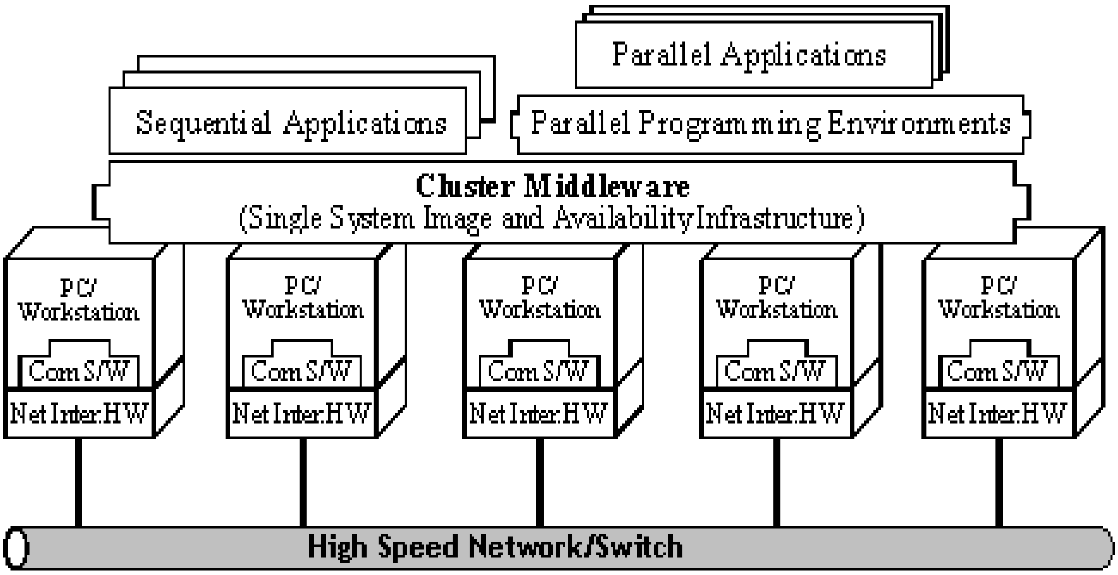

Figure 1: Typical Cluster Architecture. [4]

A generic architecture of a cluster computer is shown in Figure 1. A node of the cluster can be a single or multiprocessor computer, such as a PC, workstation, or symmetric multiprocessor (SMP). Each node has its own memory, I/O devices and operating system. A cluster can be in a single cabinet, or the nodes can be physically separated and connected via a LAN. Typically, a cluster will appear as a single system to users and applications.

This communication intends to present prospective topics in cluster computing along with some information sources (books, software, and materials on the Web).

7 Prospective Topics for Teaching

With the progress of research on cluster computing, more and more universities have begun to offer various courses covering cluster computing. A wide variety of content can be taught in these courses. Because of this, a difficulty that arises is the selection of appropriate course material.

The selection is complicated by the fact that some content in cluster computing is also covered by other courses such as operating systems, networking, or computer architecture. In addition, the background of students enrolled in cluster computing courses varies.

Cluster computing provides an inexpensive computing resource to educational institutions. Colleges and universities need not invest millions of dollars to buy parallel computers for the purpose of teaching "parallel computing". A single faculty member can build a small cluster from student lab computers, obtain free software from the Web, and use the cluster to teach parallel computing. Many universities all over the world, including those in developing countries, have used clusters as a platform for high performance computing.

Many resources are available for teaching cluster computing. For example, the IEEE Computer Society Task Force on Cluster Computing (TFCC)[1] provides online educational resources.

Even with all of the available resources for cluster education, it is difficult to design a good course that covers a reasonable subset of topics of cluster computing. The first difficulty has to do with the diverse set of topics that cluster computing entails.

These aspects of cluster computing make the development of good course material difficult, even though it must be considered three areas:

- System architecture;
- Parallel programming; and
- Algorithms and applications.

Instructors may choose selected units in each of the three topical areas and develop their own syllabus to meet course objectives. For example, a full course can be taught on system architecture for core computer science students. Or, a course on parallel programming could contain a brief coverage of system architecture and then devote the majority of time to programming methods. Other combinations are also possible.

8 A Course Structure

A cluster computing course can focus on a number of topics, including system architecture, programming environments and languages, the design of algorithms, and applications. One option is to conduct a single course on cluster computing that comprises selected units from different course components. For example, instructors can pick complementary topics from system architecture and parallel programming to develop a course.

8.1 System Architecture

The system architecture course can be divided into four units: introduction, cluster building blocks, system-level cluster middleware (focusing on single system image and high availability infrastructure), and projects. Among these units, it is advisable to dedicate the largest amount of time (more than 50%) to system-level middleware.

Many different computer architectures supporting high performance computing have emerged. These include: vector processors, Massively Parallel Processors (MPP), Symmetric Multiprocessors (SMP), Cache-Coherent Non-Uniform Memory Access (CCNUMA), distributed systems, and clusters.

Clusters are composed of commodity hardware and software components. The hardware components include standalone computers (nodes) and networks. Cluster nodes can be PCs, workstations, and SMP’s.

[1] - http://www.ieeetfcc.org/.

Networks used for interconnecting cluster nodes can be local area networks such as Ethernet and Fast Ethernet, system area networks such as Myrinet and Quadrics switches, or upcoming InfiniBand communication. Various operating systems, including Linux, Solaris, and Windows, can be used for managing node resources. The communication software can be based on standard TCP/IP or user-level messaging layers such as VIA[2].

System-level middleware offers Single System Image (SSI) and high availability infrastructure for processes, memory, storage, I/O, and networking. The single system image illusion can be implemented using the hardware or software infrastructure. A modular architecture for SSI allows the use of services provided by lower level layers to be used for the implementation of higher-level services.

Absorbing the entire course's conceptual material is impossible without hands-on experience of some aspect of cluster systems. Fortunately, several cluster-based software systems are freely available for download (with source code), including Linux, VIA, PBS, Condor, MPI, PVM, GFS, GLunix, and MOSIX[3]. Students can explore these components by changing some of the policies used in these systems. Some of the projects that can be explored include:

- Build a low-cost cluster using PC's, Linux and Ethernet Install PVM and MPI on a cluster
- Evaluate various job management systems
- Develop tools for system administration
- Develop a simple job scheduler
- Implement a standard user level communication layer based on VIA
- Develop cluster monitoring tools
- Develop share-based scheduling policy and implement in systems such as PBS
- Develop a computational economy-based policy for scheduling and implement it for PBS
- Develop a Web based job submission system
- Develop parallel Unix tools

Students can also identify deficiencies and limitations of existing systems and develop new solutions and policies to overcome them.

8.2 Parallel programming

A course in cluster computing that focuses on programming can provide students with an abundance of practical experience with clusters. The tools that are required to teach these topics are generally available on most campuses, computer science students generally have the background required at the senior level to study cluster programming, and much tutorial material is available on-line in various locations. Thus, the course can cover a range of topics without requiring the students to purchase a large number of expensive textbooks. It may be possible to teach a course on cluster programming with fewer prerequisites than is required for a course covering more advanced architectural topics.

At a minimum, students should have been exposed to data structures, basic algorithms, and computer organization prior to the units in a course on cluster programming. A course on programming can be divided into four major units: (1) Shared Memory Programming, (2) Message Passing Primitives, (3) Parallel Programming Using MPI and (4) Application-Level Middleware. These units are largely independent of one another, although the prerequisites for each unit may vary.

[2] - Virtual Interface Architecture, http://www.viarch.org/.
[3] - MOSIX – http://www.mosix.cs.huji.ac.il/.

Several languages are available to teach shared memory programming. The most accessible include:

- C or C++ using the pthreads (POSIX Threads) library on Linux or Unix, or the threads library on Solaris.
- Java threads. An instructor using Java for shared memory programming may want to spend some time discussing programming techniques for Java threads. Depending on how much time the instructor wants to spend on language skills, the students do not have to have prior experience in Java to teach this unit in Java.

Although new high-performance protocols are available for cluster computing, some instructors may want to provide students with a brief introduction to message-passing programs using the BSD Sockets interface to Transmission Control Protocol/Internet Protocol (TCP/IP) before introducing more complicated parallel programming with distributed memory programming tools. If students have already had a course in data communications or computer networks then this unit should be skipped.

An introduction to distributed memory programming using a standard tool such as Message Passing Interface (MPI) is basic to cluster computing. Current versions of MPI generally assume that programs will be written in C, or C++. However, Java-based versions of MPI are becoming available.

The resources for students for this unit include a networked cluster of computers with MPI installed. Setting up a cluster the first time can require some effort, but once the cluster is set up little to no maintenance is required on it throughout the semester.

Application-level middleware is the layer of software between the operating system and applications. Middleware provides various services required by an application to function correctly. A course in cluster programming can include some coverage of middleware tools such as CORBA, Remote Procedure Call (RPC) or Java Remote Method Invocation (RMI).

9 Conclusions

The large development of new technologies and the need of learning flexibility have favoured the concept of e-learning.

It is a general consensus that e-learning will continue to grow, as more and more individuals feel the need to add to their current education and there is less time to attend the traditional learning system.

Moreover, the demand for e-learning will bring forth a lot of investment, both in terms of money and effort spent in turning e-contents as professionally built as their off-line counterparts. As the quality of content continues to improve, so will people's expectations.

Both e-learning and clusters are two important technological areas with great expansion. Cluster computing ties together systems, communications, architecture, programming, applications, and algorithms. The nature of cluster computing allows students to tie together material from a number of different courses in their curriculum to provide a more or less complete experience in the field of computer architecture.

Our need for computational resources in all fields of science, engineering and commerce far weigh our ability to fulfil these needs.

The usage of clusters of computers is, perhaps, one of most promising means by which we can bridge the gap between our needs and the available resources.

References

[1] Dunbar A., Genesis of an Online Course, University of Connecticut (2001)

[2] Apon A., Buyya R., Jin H., and Mache J., Cluster Computing in the Classroom: Topics, Guidelines and Experiences

[3] Pfister G., In Search of Clusters, Prentice Hall PTR, 2nd Edition, NJ, (1998)

[4] Buyya R., High Performance Cluster Computing: Systems and Architectures, Prentice Hall, (1999)

[5] Demb A., The Intellectual Supermarket, Educause Review, (July/August 2002)

[6] Bakker M., and Buyya R., Cluster Computing at a Glance, Prentice Hall