Computer Clustering Using Home Network

V. Rajaravivarma
Computer Electronics and Graphics Technology
School of Technology, Central Connecticut State University
1615 Stanley Street, New Britain, CT 06050
RajaravivarmaV@ccsu.edu

Abstract

In the last few years computers have come to dominate almost every aspect of our life. It is hard to imagine the modern office or any other workplace without a computer. Moreover, more and more people use computers at home to work or just for fun. Because of their popularity people continue to find new places for their use. Although computers increase in sophistication every six months, the software and problems that people want to solve in using them have become increasingly complex and demand more computer power. The simplest solution would be to build a more powerful computer or supercomputer, depending on the scale of the problem. However, this is not an available option for two reasons: the limits of technology and the very high cost of this kind of computer. To overcome these problems, programmers had the idea to connect computers together using a network, creating a computer cluster. This paper will discuss a few clustering solutions/technologies that might be used by home computers, increasing their calculation power and graphic capabilities, such as Mosic/openMosci, OpenSSI, Nvidia SLI, BOINC and ClusterKnoppix.

Introduction

The concept of clustering is very simple. Basically, instead of building a more powerful computer or processor, which is very expensive and sometimes not possible with regard to technological development, the same speed and reliability can be achieved by connecting many low-cost computers using special software and hardware. This group of computers works together to solve the current problem and the main node runs special software that is designed to divide the current problem into subproblems. Next each sub problem is distributed to the other nodes. All the nodes simultaneously solve their sub problem. When the work is done each node sends its result to the main node where all the pieces are put together. Nodes are connected to each other, usually by network. It is very important that all computers in the group are seen by the application that is currently running as a single system (one computer). There are two types of clusters:

- High–availability clusters that provide redundancy to the system. In case one component fails, other nodes can still work, thus eliminating a single point of failure. We can see these kinds of clusters doubling the servers that work for financial institutions like banks.
- High-performance clusters. As its name implies these clusters are used to improve the performance of the system by splitting a task between all the nodes in the cluster. They are commonly used in scientific tests and simulations, as well as in graphic rendering.

In practice, however, many clusters work by providing an application with better performance and redundancy at the same time. A good example of a service that needs both types of clusters at one time is the global airline reservation system "Sabre" that handles 35% of all worldwide travel reservations and processes 15,000 transactions per second.

Cluster computing is strongly connected to networking and engineering technologies. During the exchange of information the key is speed. Cheaper and faster technologies have allowed for the development of cluster systems that are relatively cheap, very efficient and easy to implement, making them more attractive for public institutions, private companies and private users. Moreover, the majority of software applications that provide cluster service are available as open source software and can by downloaded from the internet.

Clustering technique advantages and disadvantages

There are many readily available literatures describing how to build a cluster computer, depending on the operating system of interest [1]-[3]. In general, a cluster computer consists of several nodes (PCs or workstations), each containing one or more processors, memory and additional peripheral devices connected by a network that allows data to be transferred between the nodes. Depending on its cost, performance, and reliability, different network technologies can be selected, including Fast Ethernet or Gigabit Ethernet [4]. A network of PCs does not constitute a cluster until all PCs are configured to work together and act as a team to process a single task. This requires initializing the software environment on the cluster, which allows running applications using parallel programming. There are a number of software tools supporting cluster computing such as PVM (Parallel Virtual Machine) [5] and more recently MPI (Message Passing Interface) [6]. Using standard benchmarks, such as High-performance LINPACK, efficiency and performance of different cluster systems can also be measured and compared [6]-[8].

The main concern that led us was constant difficulties that students were experiencing during the course of loading and installing different software tools to configure the cluster. Even some of the step-by-step procedures proved to be cumbersome and time consuming to implement for many students. Furthermore, the examples included in most existing software toolkits often provide no insight as to what is happening and how each node is contributing to the overall execution of an application, where the bottlenecks are, and how system constraints and modifications can impact the performance. Considering

all these shortcomings we believe, for the purpose of home networking, our approach serves as an excellent tool.

Hardware and Software

Creating computer clusters in a home environment is very easy and does not require any specialized knowledge. There are many ways to implement a computer cluster, depending on the software and hardware chosen. However, in all the scenarios we need the following equipment:

- At least two computers with network interface cards (NIC)
- Some kind of medium to exchange information, in our case UTP cable cat.5; however, in theory it might be any network cable.
- For more than two computers we need a device like a hub or switch to allow all the nodes to communicate with each other.
- Special software capable of creating a cluster from many computers and managing exchanged information.

The most popular, and so far the only ones available for free, are clusters based on the Linux operating system and special cluster server/services, for example, OpenMosic or OpenSSI. Microsoft Cluster Server (MSCS), however, is a clustering service available for Microsoft Windows Server 2003.

Linux Clustering Technologies

The popularity of Linux is due to its high reliability and free access. Many cluster solutions are available as free servers/services to download from the internet. It is relatively easy to build a cluster in a home environment based on this software. The most popular and easy to use are OpenSSI and OpenMosix. For presentation purposes and more, the most user-friendly operating system with a built-in cluster service is ClusterKnoppix.

OpenSSI and OpenMosix

OpenSSI and OpenMosix are both a single system image (SSI) [9] clustering solution for Linux. With Linux and one of those servers installed on all workstations we can create a cluster and distribute work across the nodes. It is recommended to use those systems in computer simulations and complex scientific calculations as well as 3D animations and rendering. The only condition for the application to run on the cluster is to allow the calculating of individual subproblems in independent processes. This way the cluster server can distribute those processes to other nodes. The process that distributes subproblems is fully automatic. Both those servers also automatically detect the number of nodes in the cluster and consist of mechanisms which monitor the utilization of each node. If there is a need for more system resources, the server automatically manages the processes and sends them to the best available node. This mechanism provides what is called “load balancing”. In other words, load balancing allows for the processes’

migration from slower to faster nodes and from nodes where utilization has reached 100% to other less loaded nodes. [10]

After installation of the cluster server we can easily check if the cluster works properly and if it meets our expectations. To do this we can perform a few simple tests. The specifications of our test equipment and conditions are as follows:

- The test was carried out on 16 computers.
- Each computer had Linux Slack ware 9.1 operating system and OpenMosix server installed.
- Each computer was equipped with INTEL Pentium IV 2.6 GHz, processor with 1 MB of cache.
- Each computer was given 256 MB of DDR SDRAM.
- The network between computers was running on 100 Mbps using standard Ethernet 10/100 Mbps adapters, cables and switch.

The following is an example of the “awk” performance test that is recommended by the creators of the OpenMosix:

```
awk 'BEGIN {for(i=0;i<10000;i++)for(j=0;j<1000;j++);}' &
awk 'BEGIN {for(i=0;i<10000;i++)for(j=0;j<1000;j++);}' &
awk 'BEGIN {for(i=0;i<10000;i++)for(j=0;j<1000;j++);}' &
awk 'BEGIN {for(i=0;i<10000;i++)for(j=0;j<1000;j++);}' &
awk 'BEGIN {for(i=0;i<10000;i++)for(j=0;j<1000;j++);}' &
awk 'BEGIN {for(i=0;i<10000;i++)for(j=0;j<1000;j++);}' &
awk 'BEGIN {for(i=0;i<10000;i++)for(j=0;j<1000;j++);}' &
awk 'BEGIN {for(i=0;i<10000;i++)for(j=0;j<1000;j++);}' &
awk 'BEGIN {for(i=0;i<10000;i++)for(j=0;j<1000;j++);}' &
awk 'BEGIN {for(i=0;i<10000;i++)for(j=0;j<1000;j++);}' &
awk 'BEGIN {for(i=0;i<10000;i++)for(j=0;j<1000;j++);}' &
awk 'BEGIN {for(i=0;i<10000;i++)for(j=0;j<1000;j++);}' &
awk 'BEGIN {for(i=0;i<10000;i++)for(j=0;j<1000;j++);}' &
awk 'BEGIN {for(i=0;i<10000;i++)for(j=0;j<1000;j++);}' &
awk 'BEGIN {for(i=0;i<10000;i++)for(j=0;j<1000;j++);}' &
```

The test as we can see was launched in 16 processes. The following are the results of the test:

Number of nodes	**Total calculation time**
16	31 sec
8	59 sec
4	1 min 43 sec
2	1 min 20 sec
1	1 min 57 sec

The other performance test that we can run is Blender. Blender is an open source program to create 3D computer graphics. During the test the cluster is going to calculate 1440 frames of computer animation. This kind of test is very well balanced between nodes because each frame might be calculated independently on each node. Here are the results:

Number of nodes	Number of frames	Total calculation time
16	1440	6 min 9 sec
8	1440	7 min 47 sec
4	1440	14 min 5 sec
2	1440	27 min 36 sec
1	1440	49 min 8 sec

As we can see the two tests show us that the cluster shares processes between nodes which usually results in decreasing calculation time. The calculation time might increase because of unequal load balancing. [11]

It is not recommended to use clusters in calculations that use a lot of hard drive memory because all the information must be sent through the network to other nodes and the loss of time causes significant delay. OpenMosix does not require a hard drive space for installation. It can be launched and run from the CDROM drive.

ClusterKnopix

Another good example of clustering technologies based on Linux system is ClusterKnoppix. ClusterKnoppix is basically the Knoppix Linux operating system that uses OpenMosix's kernel. Knoppix boots completely from CDROM and is based on Debian, with a KDE desktop environment and a lot of additional free software like OpenOffice.org, Mozilla etc. It takes 1.7 GB so it requires 1 DVD or 3 regular CDs. However, ClusterKnopix was designed and compressed to fit on a 700MB CD. We can think about it as a demonstrative version of combined OpenMosic and Knoppix. The main features of ClusterKnoppix are:

- OpenMosix terminal server that allows us to boot other computers through the network using tftp. The only condition for network booting is to have NIC able to boot from LAN.
- No hard disk or CD ROM drive is needed for the clients. It can be run without having to be installed on the hard drive and without breaking the existing operating system settings etc.
- All nodes are auto-detected by the OpenMosix auto discovery process. The ClusterKnoppix author claims no configuration is needed; however, there is one very important point during the startup of the OpenMosix terminal server. We will be asked about the other workstation NIC's driver and we need to specify it correctly. If not, the distribution of the system cannot be done.
- New nodes automatically join the cluster.

The following is the cluster test demo that was performed by this author on his two computers. Computers were connected by network cable using a Cisco 4-port switch which, using DHCP, leases ip addresses automatically. Despite the fact that the specifications of the two workstations were totally different, including different models of processors, amount and kind of RAM and chipsets, setting up the cluster server was easy, quick and smooth.

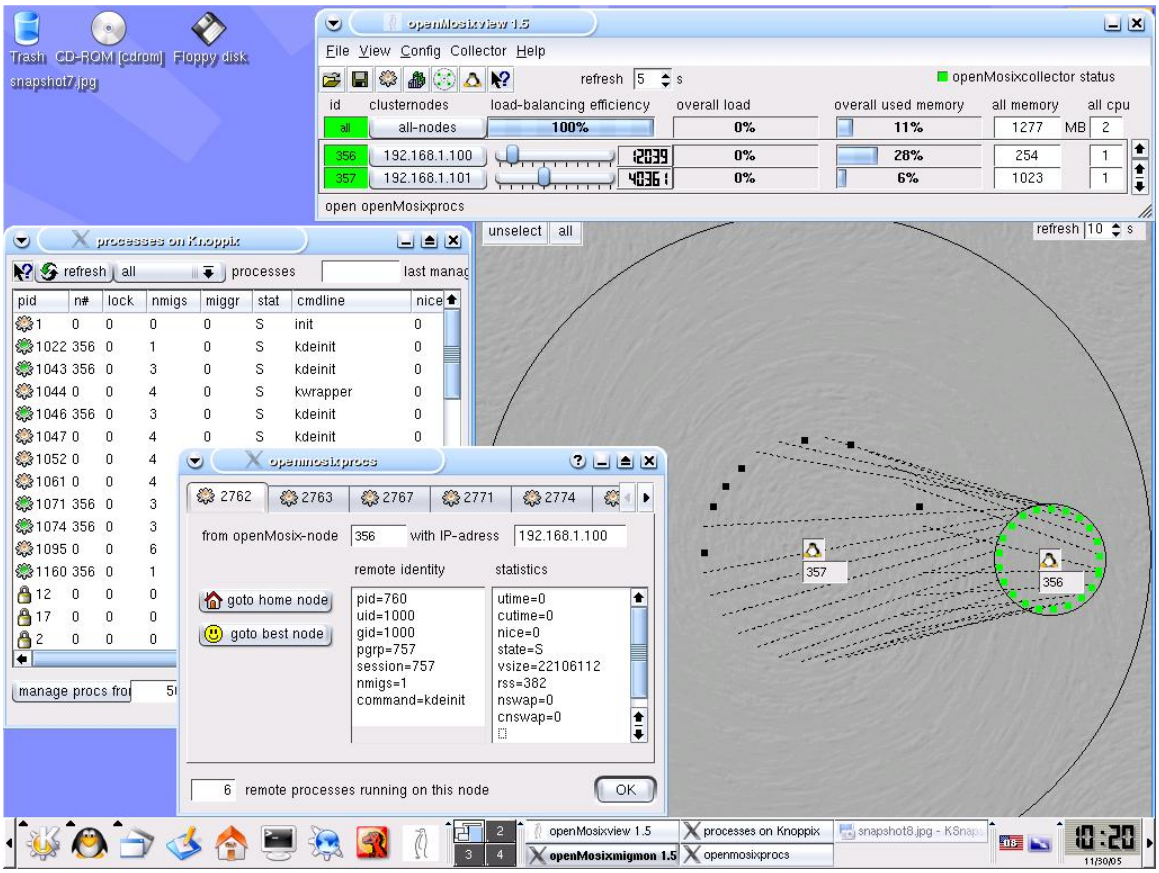

Fig 1. Screenshot from the workstation running ClusterKnoppix.

As we can see at the top of the screen OpenMosix server 1.5 window is opened on workstation no.1. This view gives us information about the total number of CPUs, RAM currently being used and load balancing efficiency between nodes. In this view we can see two penguins on a gray background. The two penguins symbolize two working computers. The penguin in the middle always represents the current workstation. Around the penguin we can see a circle of black dots. Each dot is actually a process that is running on the workstation. We can simply drag and drop processes from one workstation to another. The green dots around the workstation no.2 represent processes that migrated from workstation no.1 to workstation no.2. The same view might be open on workstation no.2 and its processes can also migrate to the first one. The other two open windows show us detailed information about all the processes.

ClusterKnoppix is very easy to install and manage. It is a great tool that can be used for demonstrative purposes, automatically creating a real, fully operational cluster. Its free accessibility makes it very attractive for learning, discovery and testing clustering technologies for everybody. [12]

BOINC

BOINC is a free program for anyone who wants to share his computer resources and computational power.[1] BOINC relies on volunteers and its projects are considered to be nonprofit. BOINC works by distributing subproblems to computers around the world. When the work is done its result is collected by the server and put together. The advantages of this kind of solution are quite obvious. Even the most advanced computer cluster is limited to a certain number of nodes based on the financial capabilities of the institution. BOINC is theoretically unlimited in number of nodes. The BOINC server can have as many nodes as can be obtained from the internet. The only condition is that the problem must by dividable into so many subproblems for all the nodes. There might be many categories of projects. BOINC arose from SETI@home project which looks for extraterrestrial intelligence by using connected computers to analyze radio telescope signals from around the globe. Soon after the great success of SETI@home, its creators understood that they might be able to use the same concept to solve or analyze many other problems – this was the beginning of the BOINC. [13]

Windows vs. Linux

As mentioned before, most cluster solutions are design to work with Linux. Microsoft, however, intends to increase its focus on Windows in high performance computing but has yet to make a measurable presence in the market. Microsoft Cluster Server (MSCS) is a clustering service available for Windows Server 2003 that provides failover and availability of applications, as well as performance improvement for up to 8 nodes. [14] In the future, however, Microsoft is going to release a cluster server that can manage fewer than 200 nodes. This solution is addressed to scientists who want to test their simulations or models at once, without having to wait for the supercomputer or big cluster. According to Microsoft, it will be possible using their Windows cluster servers and then if the results are satisfactory, the same experiment or problem can be analyzed by a big Linux cluster or supercomputer.
The Microsoft cluster server is going to be easy to implement and most scientific institutions run Windows anyway. Moreover, Microsoft is going to provide helpdesk service for all those clusters, helping scientists to set it up and run it. In my opinion, Microsoft is on the right path to achieving success in this area. [15]

SLI

Many cluster servers are used by graphic engineers for animation rendering. Nvidia's SLI technology is not exactly a clustering technology but I included it in this paper because in my opinion it imitates a computer cluster that is used for graphic purposes. SLI might be used in a performance testing scenario. All we have to do is to download the graphic demo that requires high computing power and test it on the computer. Than we can add another graphic card to the system and connect them via SLI. SLI basically clusters two graphic cards (two graphic processors and its memory) in one system, increasing the system rendering and graphic capabilities up to two times. What differentiates SLI from

[1] BOINC stays for Berkley Open Infrastructure for Network Computing.

symmetric multiprocessing is that the graphic cards, in contrast to the processors on one motherboard, do not share memory. Each card is managed by a special MCP chip on the motherboard. [16]

Conclusion

In today's computer-dependant world, clustering has become an essential tool. All projects which require high computer power but for financial reasons do not have access to it might be improved or developed using relatively cheap and easy-to-implement clustering. There is a lot of free software that provides cluster services to download from the internet as well as online help in case of any problems.

Bibliography

[1] Windows Compute Cluster Server 2003 Product; updated November 15, 2005; available online at Overview http://www.microsoft.com/windowsserver2003/ccs/overview.mspx

[2] Karl Kopper, "*The Linux Enterprise Cluster: Build a Highly Available Cluster with Commodity Hardware and Free Software*", Oreilly & Associates Inc., 2005.

[3] How to Build a Beowulf - by Thomas Sterling, Paul Angelino, Don Becker, Jan Lindheim, & John Salmon http://www.cacr.caltech.edu/beowulf/tutorial/tutorial.html

[4] Computer Networks and Internets, Dougles E. Comer Prentice Hall, Inc. 1999. 3-Designing and Building Parallel Programs (online), Ion Foster

[5] Geist, A., A. Beguelin, J. Dongarra, W. Jiang, R. Manchek and V. Sundaram (1994), *PVM: Parallel Vir- tual Machine*, The MIT Press: Cambridge, Massachusetts

[6] Pacheco, P. (1997), *Parallel Programming with MPI*, Morgan Kaufmann Publishers Inc.: San Francisco, California.

[7] Jack J. Dongarra et al. *Java Linpack Benchmark*; available at `http://www.netlib.org/benchmark/linpackjava/`

[8] LAPACK -- Linear Algebra Package available at http://www.netlib.org/lapack/ *4-How to Build Cluster Computers* (online article); www.devbuilder.org/article/24

[9] SSI means that for all applications a cluster that consists of many computers is seen as one system

[10] J. Solan, High Performance Linux Clusters with OSCAR, Rocks, OpenMosix, and MPI, First Edition November 2004, for more information go to: **http://openmosix.sourceforge.net/**

[11] For more information go to: **http://www.klastry.org**

[12] For more information go to:**http://bofh.be/clusterknoppix/** and **http://www.knoppix.net/**

[13] For more information go to: http://boinc.berkeley.edu/

[14] For more information go to:

http://www.microsoft.com/windowsserver2003/evaluation/overview/technologies/clustering.mspx

[15] Aaron Ricadela, *Supersized Challenge* in Information Week, Nov. 21,2005,36-42.

[16] For more information go to **http://www.nvidia.com**

Biography

Veeramuthu Rajaravivarma
V. Rajaravivarma is currently with the Computer Electronics and Graphics Technology department at Central Connecticut State University, New Britain, CT. Previously, he was with Tennessee State University, Morehead State University, and North Carolina A&T State University. Dr. Rajaravivarma teaches a variety of courses to undergraduate and graduate technology and engineering technology students. His research is in computer networking and digital signal processing