Clusters Systems based on OS

Richard Wallbrink, Marcus Pallinger

School of Computer Science and Software Engineering
Monash University
Clayton Campus, Melbourne, Australia
Email: raw, marcusp@csse.monash.edu.au

Abstract

There are some operating systems that have been built around the idea of clustering computers together. These operating systems tend to be written with specific high end computers in mind, or especially written with a certain purpose in mind.

Other developers have taken a different approach. They have added clustering abilities onto existing operating systems. This allows a cluster to exist using cheap commodity hardware and existing operating systems. This report will look into the clustering ability of various commonly used Operating Systems.

1 Introduction

Clustering on commodity hardware has lead to a cheap way to produce parallel computers, at the expense of slower communication between processors. Clusters can, thanks to the addition of modifications to common operating systems, be built out of commodity hardware, rather than purpose built computers designed for clustering.

There are many choices in what hardware, Operating system and clustering add-on can be used to build the cluster. The hardware and operating systems will probably be first decided upon, based on availability, funds and abilities of the operating system.

There are several different types of clusters. They include failover, load balancing and parallel computing. Failover clusters are designed so that there is always a second machine (and possibly more) that waits until the primary machine in the cluster fails. When the primary machine fails, the secondary assumes it's duties in a way transparent to the users. Load balancing is the sharing of work between nodes in a cluster to ensure that no nodes are overloaded. Load balancing clusters can either load balance services or processes. Service load balanced clusters deal with several services, eg. web servers, running on all nodes of the cluster, and requests being assigned to individual nodes. Process based load balancing is where actual processes running on nodes in the cluster are migrated from one node to another, as nodes become overloaded. Parallel computing clusters are designed to allow programs to be especially written, allowing them to run at the same time over multiple nodes in the cluster.

2 Works on this area

- Linux
  - Beowulf http://www.beowulf.org/

  - Mosix http://www.mosix.org/
  - Linux High availability project http://linux-ha.org/

- Windows NT
  - MSCS http://www.microsoft.com/ntserver/ntserverenterprise /techdetails/prodarch/clustarchit.asp
  - HPVM http://www-csag.ucsd.edu/projects/clusters.html

- Mac OS
  - Appleseed http://exodus.physics.ucla.edu/appleseed

- Solaris
  - Sun Cluster http://www.sun.com/software/white-papers/wp-sunclusters /sunclusterswp.pdf
  - Sun Cluster http://www.sun.com/software/white-papers/wp-sunclusters /sunclusterswp.pdf
  - Solaris MC http://www.sun.com/research/technical-reports/ 1995/smli_tr-95-48.pdf
  - Berkley NOW http://now.cs.berkeley.edu

3 Linux Clustering

As Linux is a freely available operating system, both in terms of money, and in terms of source code availability, there have been several projects that have found it easy to add clustering abilities. The Linux operating system has been designed to run well on off-the-shelf equipment, and has fairly stable Symmetric Multi Processor (SMP) support, which some clustering implementations have built upon.

Beowulf

The Beowulf project started at NASA's Center of Excellence in Space Data and Information Sciences. The first Beowulf cluster was created in 1994[2], and people seeing the ability to build a cluster using commodity off the shelf hardware. One of the changes made to Linux was the ability to use multiple network cards to convey data between nodes of the cluster. The nodes in a Beowulf cluster are dedicated to only being nodes in a beowulf cluster.

Beowulf clusters are designed for parallel computations, for example simulations. To allow clusters to communicate, two Application Programming Interfaces (API) are supplied, MPI and PVM. This suits Beowulf clusters for any tasks that require communication and parallelism.

Mosix

Mosix clusters take a different approach to clustering than Beowulf. Mosix is more of a load balancing system, by automatically allocating processes across the machines in the cluster. It is essentially an emulation of SMP over multiple nodes in a cluster.

The advantage of Mosix's approach is that it allows programs to be run on clusters without any modification or re-compiling. Processes are executed on nodes of the cluster, on basis of load[7]. Mosix is supplied as a set of patches to the Linux kernel, and a set of utilities.

Applications to which Mosix is geared are:

- CPU bound Processes. that have little Inter Process Communications.
- Multi-User Time-sharing systems. Mosix clusters can give the appearance of a large computer. The Computer Science department of the Hebrew University (The people that developed Mosix) have an 8 node cluster of 8 medium powered computers that is used by 650 students, who do heavy computations[4].

Mosix is not designed to handle tasks that require a lot of communication between processes. The designers of Mosix recommend that MPI or PVM are used instead.

High Availability Linux Project

The High Availability Linux Project has another approach to clustering. The aim of the project is to create a cluster of servers, where one server will automatically take over the workload of another server if it goes down. This type of cluster is not aimed at parallel processing, or load sharing. It's one goal is to have a pool of computers to take over from a server if it stops working.

There are several different ways of implementing the cluster[3]:

- Idle Standby. There is a Master computer and a cluster of standby computers.. As soon as the Master goes down, a computer form the cluster, which has the highest priority then takes over. If a machine with a higher priority is inserted into the cluster, it will take over as the master (causing a brief service outage)
- Rotating Standby. This implementation is similar to the Idle Standby implementation, but the first machine in the cluster is the active one. When it goes down the second node will become the active node, and when the original node is re-inserted into the cluster, it becomes the standby node.
- Simple Failover. This strategy is similar to the Idle Standby strategy, but rather than having the second computer idle, the second computer will be a different server, for example a development computer, but if the master server goes down, it will take over.
- Mutual Takeover. Basically this strategy is a combination of Idle Standby and Simple Failover. Two machines are serving different applications and each is designed to take over the other's duties plus it's own if the other node goes down.

4 Cluster based on Windows NT

Clusters that use the Windows NT operating system are somewhat of a recent invention. The use of software like High Performance Virtual Machine (HPVM), developed by Andrew Chien of University of Illinois and his students, makes the use of NT clusters possible. The software enables each node of an NT cluster to communicate at a bandwidth of just less than 80 megabytes per second and a latency under 11 microseconds using Myricom's Myrinet interconnect [1]. The use of NT to make high performance clusters means that the scientists and programmers that need to use these environments have well-known programming and run-time services. This can allow for greater use of the clusters, as more people can feel confident in the utilization of them. There is a need to have NT based clusters not only for the cost but also because some researchers feel more comfortable using a platform that they are already accustomed. Another reason is that the well-known unreliability in the OS or a failure in the hardware can be rectified by using clusters, where each node can monitor each other's activities and then be able to recover from a system failure.

MSCS

For an NT workstation to be able to have effective cluster management and scheduling there must be software implemented on top of the existing operating system. Windows does have clustering software [9], this is for the Windows servers and is an extension to the Windows NT operating system. There is also an extension for the new Windows 2000 operating system [10] this improves over the previous version for NT. The software must be obtained from Microsoft. These use the Windows NT server Enterprise Edition software and so require special purchases of this software instead of the usual Windows NT software. Microsoft has put emphasis on the user interfaces of the programs. The Microsoft has two cluster technologies that have been implemented in their newest Microsoft Windows 2000 Advanced Server and DataCenter Server operating systems being of Cluster service (MSCS) and Network Load Balancing (NLB). The Windows 2000 Advanced Server and 25 Client Access Licenses cost around $6500 (Australian) and is not a cheap option for making large clusters. The cluster service provides fail-over support, with 2-node fail-over clusters in Windows 2000 Advanced Server and 4-node clusters in DataCenter Server. NLB is a service that load balances incoming IP (Internet Protocol) traffic across clusters of up to 32 nodes. These two technologies can then be used to make somewhat reliable and scalable clusters. It should be noted that the load balancing is for only up to a cluster of 32 servers. The Windows 2000 DataCenter can support as much as 16 processors and is still behind that of the High-end Unix servers, which can hold from 64 to 256 processors each [17]. This means that the Windows operating system is still behind that of the Unix servers. It should be noted that there is a category of computers that these Unix clusters, with thousands of processors, fall just short. This category is massively parallel systems, which are single servers that support hundreds or thousands of tightly joined processors.

The Windows 2000 operating system has developed a new technique called the job object. This gives a method for the grouping of processes, threads and the resources that are used, so that they can be recognized as a single entity [17]. Apart from MSCS, there are other OS support level cluster systems like Vinca Standby Server and Legato Fulltime. These systems also monitor the activity between members of the cluster, and execute tasks like replicating required system information and scrutinizing the operational status.

HPVM

The HPVM communication suite is implemented on the Windows NT operating system and provides 10μs latency and 90MB/s bandwidth employing Fast Messages to implement MPI, Global Arrays, Shmem Put/get[14], flexible scheduling, system monitoring, job submission, and control of jobs via a Java applet front-end which runs on any Java-enabled system [12]. The HPVM project has the goals of providing supercomputer performance on low cost off the shelf systems [8]. It also has the goal of hiding the complexities of a distributed system behind a clean interface, which is accomplished by the Java front end. HPVM has software components like MPI, SHMEM and Global Arrays that allow it to be competitive with dedicated MPP systems. The HPVM system uses a low-latency and high-bandwidth communications protocol known as Fast Messages (FM), which is based on Berkeley AM. Reliable and ordered packet delivery and control over communication work scheduling is all guaranteed by FM. It also has functions for the sending and receiving of long and short messages through the network.

The HPVM III is a general-purpose machine that was developed for the NCSA and now in use at the NCSA in Urbana. The machine has 96 nodes and 192 300 MHz Pentium II processors. The performance of the machine has been found to be within a factor of two to four of that of the SGI Origin 2000 and the Cray T3E. The main benefits are that the cost of the machine is a lot less and that there are useful tools, for example remote administration of services. The total cost of the computer is $200,000 but the performance is that of a multimillion-dollar machine, which is proof that the use of NT clusters can be cost effective in the making of a machine for supercomputing applications [11]. The cluster can be accessed from the Internet and a Terminal Server session displays the NT desktop and uses the computing power of the server to run applications. This cluster is the first to demonstrate that the NT operating system is good enough to be able to be used in the making of machines with supercomputing power. Its main problem is in efficiency in the high-performance storage and thus this area has been looked at to change the performance of the cluster. The developers of the system have been looking at fiber channel to improve the performance level of the cluster.

Many applications have been developed for clustering the Unix operating system and could be ported over to the Windows NT system. This is one of the major delays in the time that was taken to develop the HPVM system for Windows NT. Compaq has developed servers that use the MSCS system to enable clustering on with the server [17]. It has developed better cluster management tools and has offered a degree of assurance that the system will have 99.5 percent availability. This means that the down times for the server are approximately 40 hours a year, but this is below that of Unix systems which offer 99.9 to 99.999 percent availability. This is almost matched by the development by HP which uses Marathon interface cards and have the status of servers monitored by a remote management workstation, to achieve 99.99

percent availability.

5 Cluster based on Mac OS

There are only a few clusters that have the Mac OS as there operating system these are: the AppleSeed cluster, the D'Artagnan Cluster at the Department of Physics and Astronomy at York University in Toronto, the University of Wisconsin High Performance Computing Cluster, a Cluster of Four G3's at Universität Dortmund, the OAN Cluster at Observatorio Astronomico Nacional of the Universidad Nacional de Colombia, the MRI Data Processing Cluster at the University of Maryland School of Medicine and

The Lovecraft Cluster [5].

Performance of the Mac clusters

These are all clusters that have been developed using the MacOS and are currently in use. The Appleseed cluster is one that has 22 Apple Macintosh and G3 and G4 computers[13]. The performance of this cluster has been compared to that of other clusters, and it has come out fairly good. A cluster of 4 Macintoshes has the same power as that of a 4 processor Cray Y-MP and the same with a Cray T3E-900 even when both have 8 processors. The performance of even only one node out does that of many other systems, so for the cost of the systems, the resulting computer power is quite impressive.

Construction of a Mac cluster

The software for the Mac has to be specifically written, with MPI on Mac being called MacMPI. This is a partial implementation of MPI with only 34 subroutines and relies on the Program-to-Program Communications (PPC) Toolbox to do the other commands of normal MPI. MacMPI_IP is a improved version that uses a TCP/IP implementation of Open Transport (a protocol-independent communications library). This version of MPI is a lot faster and can be found at the site for the Appleseed cluster [13]. The setup of the Macintosh cluster is very simple with the only need being Category 5 Ethernet cables with RJ-45 jacks and a Fast Ethernet Switch with enough ports for each Macintosh. The next step is then connect one end of each cable to the Ethernet jack on each Mac and the other end to a port on the switch. One of the main benefits of the MacOS cluster is that it is very easy to set up, with it being virtually plug and play. The MacOS cluster is also reported to have almost linear speedup using multiple processors [15]. The administration of the cluster is also very good, as it is the same as the normal administration of an Apple network, in that there is virtually none needed.

6 Solaris

Sun Clusters

Sun Microsystems have released Sun Clusters. Sun Clusters is a clustering environment built upon their Solaris Operating system. Sun Clusters provides several clustering features.

Solstice High Availability

Solstice is basically Sun's Failover cluster software[6]. It allows two servers to cover for each other, in case one of them goes down. This cluster configuration allows both the primary and the secondary servers to be in use as opposed to being idle.

Sun Enterprise Cluster Server

Sun Enterprise Cluster Server allows for large capacity and high availability applications[6]. It is a platform that supports applications that can perform analysis of data in parallel, without needing much communication, for example databases running on two nodes, but sharing one set of data. Vendors producing Parallel databases include Oracle and Informix[6].

High-Performance Computing

This cluster style supports parallel programming through APIs like PMI and PVM. It runs programs that are designed to be parallel with lots of communication between nodes.

Sun's plan for the future

Sun has a project named Full Moon[6]. The ultimate goal of this project is to provide simple and effective clustering on Solaris, as well as supporting Single System Image. Single System Image is where the cluster looks like a normal computer, rather than a series of clustered nodes. There are several phases in this project, that include the development of a cluster file system, simple management of the cluster, and the merging of Sun's clusters into one system.

The Sun Cluster software will eventually be able to detect a failure in a node, and migrate all processes off it to a fully functioning node. This is done in order to ensure that no processes are lost if there are further problems with a failing node. This approach assumes full redundancy in the nodes (ie, dual network cards, etc.).

Solaris MC

Solaris MC was designed as a prototype system by Sun Labs in 1995. It was built on top of Solaris 2.5, allowing the developers to use the existing Solaris ABI and API. This allows existing applications and device drivers to continue to be used without change. The communication between nodes in the cluster was done in an object oriented way. It has the down side of not supporting MPI or PVM, which most modern parallel computing clusters support. This makes it hard to port applications between Solaris MC and other cluster operating systems.

Part of Solaris MC's Single System Image is the globalisation of its process management. All process operations are global across all nodes in the cluster. This includes access to the /proc file system, allowing Unix commands, such as ps, to work across the cluster. Sun Labs implemented a special file system for Solaris MC, allowing distributed storage of data.

Berkley NOW

Berkley NOW is implemented through a layer called Global Layer Unix (GLUnix) [8]. GLUnix is implemented in user space (ie. running with normal processes, as opposed to running as part of the operating system kernel) , allowing it to be portable across different flavors of Unix. It was, however initially implemented on Solaris. The various cluster APIs, such as MPI, HPF and so on all communicate with the GLUnix Layer. Messages are passed between nodes using Active Messages. Network RAM is used to allow free resources on idle machines to be used for paging on busy machines, with any machine being able to page to any other machine. Like Solaris MC, Berkley NOW also uses a distributed file system, called xFS, allowing redundant file serving.

7 Comparisons between some clusters

NT compared to Linux

Performance of the NT cluster is quite good when compared to the Linux cluster [16]. The paper by Luecke, Raffin and Coyle of Iowa State University shows that the NT cluster can outperform the linux cluster (of equal processors) in the following tested areas:

- Time taken for an MPI synchronization barrier,
- Scalability due to some problems in the Linux software producing bad results,
- Right shifting where each processor gets a message of size n from its neighbor,
- A naive broadcast,
- A binary tree broadcast,
- and All to all MPI coma nd

As they concluded from these tests the NT cluster seems to be able to actually outperform that of the Linux cluster and hence should remove any doubts as to the ability of the NT operating system being able to handle clustering. The NT cluster used was one with the HPVM program allowing it to be clustered with up to 128 processors. This shows that the operating system implemented on the cluster can have an effect on the overall performance of that cluster.

MacOS compared to Linux

The MacOS has some benefits over the Linux operating system in that it has some third party software for numerical calculations that runs better on the Macintosh G3 than on the Unix workstations [13]. Another benefit is the amount of software that is available for the MacOS being quite large with programs like

Microsoft Word that aren't on Linux. The other main reason is that the Mac operating system is more user friendly and is easier to use than Linux for the novice user that might want to use the cluster. The Mac cluster is also low maintenance with only a single library and utility needed to run the cluster.

Linux is free and is therefore very suitable for low cost clusters. Another benefit is that for users that are already accustomed to the Unix operating system or even the Linux operating system, the use of Linux would be more suitable. Linux is also more adjustable to the wishes of the users. Linux clusters can run on different architectures and then make advantage of the specific hardware, where as the MacOS can't.

8 Summary and Conclusions

Each operating system has its own benefits with the Linux operating systems main benefits are that it is free and that it can run on different architectures. This allows for cheap clusters and as Linux is the most common operating system that is used today in clusters, there is a good track record for its use. The Windows operating systems on the other hand, have ease of use as their main goal, with effort made in the making the setup of the clusters as simple as possible. The HPVM software allows the cheaper version of the Windows operating system, Windows NT, to be used in clustering. This makes the cost of this form of cluster a lot cheaper than it otherwise would be and also enables Windows NT to be used as the operating system in clusters that can rival that of supercomputers. The Mac OS is very easy to use for setting up a cluster and also provides great power associated with the Macintosh computers. The Solaris Operating system is aimed at providing simple effective clustering. The ultimate goal for this operating system is to provide an all in one system that caters for all uses of clusters.

In general the Linux operating system is useful for very cheap clusters, while the NT and Mac OS clusters are more expensive they provide great ease of use. The Solaris also has a Java based console that enables easy management of the cluster. The Solaris system is designed to be parallel with lots of communication between the nodes and this is where other systems can falter. The MacOS is suitable for small groups with limited resources because of its ease of use in setting up, use and management. The best operating system for a cluster is thus the one that is best suited to the users needs.

References

[1] High performance supercomputing at mail order prices. Web site. http://www.ncsa.uiuc.edu/access.html.

[2] Introduction to the cesdis beowulf project. http://www.beowulf.org/intro.html.

[3] Linux high avalability how-to. http://metalab.unc.edu/pub/Linux/ALPHA/linux-ha/High-Availability-HOWTO-5.html.

[4] Mosix application environments.
http://www.mosix.org/txt_what_it_can_do.html.

[5] Site of other appleseed clusters. Web site.
http://exodus.physics.ucla.edu/appleseed/appleseedsites.html.

[6] Sun clusters - a white paper.
http://www.sun.com/software/white-papers/wp-sunclusters/sunclusterswp.pdf.

[7] What is mosix.
http://www.mosix.org/txt_what_is.html.

[8] *High Performance Cluster Computing: Architectures and Systems,*. Prentice Hall PTR, 1999.

[9] Windows nt server enterprise edition — technical details clustering architecture. Web Site, April 12 1999.
http://www.microsoft.com/ntserver/ntserverenterprise/techdetails/prodarch/clustarchit.asp.

[10] Introducing windows 2000 clustering technologies. Web site, February 2 2000.
http://www.microsoft.com/WINDOWS2000/library/howitworks/cluster/introcluster.asp.

[11] Christa Anderson. Big clusters solve big problems. Windows NT Magazine US Print, June 1999.

[12] A. Chien, M. Lauria, R. Pennington, M. Showerman, G. Iannello, M. Buchanan, K. Connelly, L. Giannini, G. Koenig, S. Krishnamurthy, Q. Liu, S. Pakin, and G. Sampemane. Design and evaluation of an hpvm-based windows nt supercomputer. the international journal of high-performance computing applications. *The International Journal of High-Performance Computing Applications*, 13(3):201–219, Fall 1999.

[13] Victor K. Decyk, Dean E. Dauger, and Pieter R. Kokelaar. How to build an appleseed: A parallel macintosh cluster for numerically intensive computing. Web site and in Conference.
http://exodus.physics.ucla.edu/appleseed/appleseed_report.html and International Conference on Numerical Simulation of Plasma in Banff, Alberta, Canada May 2000.

[14] Louis A. Giannini and Andrew A. Chien. A software architecture for global address space communication on clusters: Put/get on fast messages. In *Proceedings of HPC-7 '98.*

[15] Roman Koniuk and Chris Stewart. The d'artagnan cluster. web site. http://www.students.yorku.ca/ kipper/dartagnan/dart.html.

[16] Glenn R. Luecke, Bruno Raffin, and James J. Coyle. Comparing the communication performance and scalability of a linux and an nt cluster of pcs, a sgi origin 2000, an ibm sp and a cray t3e-600,. *The Journal of Performance Evaluation and Modelling for Computer Systems (PEMCS)*, March 2000.

[17] Rawn Shah. Clustering for nt: Smooth scaling? Web site, July 30 1999. http://www.windowstechedge.com/wte/wte-1999-07/wte-07-clustering.html.