Implementation of Clusters made easy

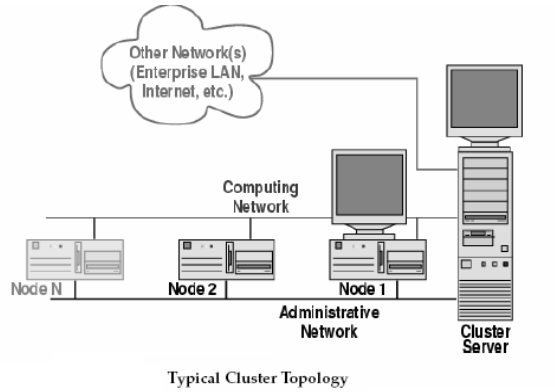
Abhinav Dhall, Vibhore Jain, Team Prerna, D.A.V. Institute of Engineering & Technology, Jalandhar. The authors are Third year B.Tech. Students and are working on Project Prerna, which revolves a supercomputer "Prerna". They can be contacted at:
abhinav_dhall@hotmail.com , vibhorejn@yahoo.co.in

Abstract
The paper revolves around supercomputing with clustering and explains in depth, as how to assemble a cluster with already available hardware in a lab. The case study is Prerna which is a Linux based cluster at D.A.V. Institute of Engineering & Technology, Jalandhar. Various aspects which are to be kept in mind while development of the system have been covered in depth. The paper is aimed from novice to experts. Reading it even enables to develop a cluster.

Introduction
The last two decades have seen steep rise in the performance and sharp decline in the cost of computer hardware. The performance has increased and so has the demand of engineering applications. Almost all type of research activities these days requires computer simulations. An obvious reason is time and money saving.

While travelling at faster-than-bullet speeds along the roads laid by Moore's Law, the semiconductor industry has hit the proverbial wall. Much to their dismay, diminishing returns and migrating electrons have made shrinking processors an uneconomical strategy. Multi-core processors, we are told, is where the future is. In a way clusters and distributed computers are already there. By parallelizing a task, a cluster offers supercomputer speeds at a fraction of the cost. It has a lot of application in places such as university and college laboratories which usually lack proper computing resources. The scenario demands more and more power but the finances have to be under control. The paper provides information on, as how to implement a cluster in a lab using the off-shelf hardware such as to achieve the best performance/price balances. The case study is Prerna - a Linux based cluster implemented by the authors in their college labs.

Now, first of all what is a cluster?. As the name suggests a **cluster** is a **network** of **computers** solving complex problems in a **parallel** fashion so as to achieve **multi-gigaflop performance**. A cluster can also be termed as a Network of Workstations performing operations together on the bases of **"Divide and conquer"** methodology. The low price of the machine is due to the use of already available hardware and mostly free open source software. It all started with the Beowulf project at NASA in 1994(1), where 16 DX4 processors were joined to form a cluster.



Typical Cluster Topology

The programming methodology on these machines is quite different from the usual sequential programming .In these parallel programming is used, in which a task is broken into multiple task and then executed simultaneously on different nodes. This decreases the execution time to T/N theoretically, on an N node system.

Basically there are 3 types of clusters **- Fail-over**, **Load-balancing** and **HIGH Performance Computing**; the most deployed ones are probably the Failover cluster and the Load balancing Cluster.

1) Fail-over Clusters consist of 2 or more network connected computers with a separate heartbeat connection between the 2 hosts. The Heartbeat connection between the 2 machines is being used to monitor whether all the services are still in use: as soon as a service on one machine breaks down the other machines try to take over.

2) With **load-balancing clusters** the concept is that when a request for say a webserver Comes in, the cluster checks which machine is the least busy and then sends the request to that machine. Actually most of the times a Load-balancing cluster is also a Fail-over cluster but with the extra load balancing functionality and often with more nodes.

3) The last variation of clustering is the **High Performance Computing Cluster**: the Machines are being configured specially to give data centers that require extreme performance what they need. Beowulf have been developed especially to give research facilities the computing speed they need. These kind of clusters also have some load-balancing features; they try to spread different processes to more machines in order to gain performance. But what it mainly comes down to in this situation is that a process is being parallelized and that routines that can be ran separately will be spread on different machines instead of having to wait till they get done one after another.

The first phase of our Project Prerna revolved around Load –Balancing clusters. Now as these have been tested now we are moving to HPCC.

Working:

There are many different technologies available for constructing a cluster. All employ different algorithms and techniques but the basic fundamental of clustering remains the same. A job which usually consists of different modules is executed at one node that may be called the control node.
Now according to the algorithms deployed by the cluster, processes are migrated to remote nodes. The processes at remote nodes may or may not communicate with each other. This depends upon the software requirements.

Perquisites:-

Before starting on how to build a cluster, the following parameters are to be analysed first:-

1) The application base and platform selection.
2) The type of network.
3) Hardware configuration

Basically point one i.e. application base decides the other two parameters namely type of network and hardware configuration. Now the most optimum use of a cluster can be made by an application having small independent module which need minimum communication in between. Multiple modules increases the parallel computing capability and least communication ensures minimum time wastage of the CPU. If the application is also required to be designed from level zero then parallel libraries such as MPI (Message passing Interface)(2), PVM (Parallel Virtual machine)(3) should be used. These libraries make the program optimised for the parallel execution. On the other hand if the application base is general i.e. if multiple applications such as MATLAB simulations, rendering jobs, SETI@Home etc. are to be executed, then the base should be openMosix. One noticeable fact is that openMosix(4) is purely Linux based. On the other hand PVM versions for windows platform are also available. OpenMosix usually speeds up all Linux based multithread applications, Java threads being an exception. MPI and PVM score when the

application is very specific and complex. Deciding the Operating System is the second major part of this point. Linux powers majority of the top 500 supercomputers in the world. This in turn automatically speaks of the stability and portability of the Linux based systems. Linux distribution specially meant for clustering are also available such clusterKnoppix(5),rocks(6),OSCAR. Coming to other Operating Systems such as Microsoft Operating Systems, they can also be employed for the job but the cost of the cluster will rise considerably. A notable point here is that clustering in Windows 2000 Advanced Server and later operating systems is very easy, but where is the parallel code?.
The only solution is the Visual C++ clustering API available from Microsoft. Once the code is made available the performance is sound. Versions of Linux such as Mandrake CLIC, Plump OS, ClusterKnoppix etc. are specially designed for clustering and are a very cost effective solution. This field has a lot of research into it and there are still tons of other solutions such as clustermatic[10],openSCE[11], Warewulf[12] etc.

Now the **network** in a cluster can be compared to the vein-artery system in a human body. A cluster constantly requires inter-process communications and process migrations. Hence the network should be considerably fast, so as to avoid congestion and improve CPU utilization. Usually 100 Mbps Ethernet connection which is generally available in Labs can work. But with the arrival of 1 Gbps LAN and network such as Myrinet and ATM networks, very fast communication can be achieved. Joining N nodes doesn't mean N times speed gain. This is due to bottlenecks such as network latencies etc. Hence it is to be noted that the speed of the network vastly affects the overall cluster output.

Coming to the hardware aspect such as the Processors, RAM , and backup RAIDS etc. More Megahertz means more performance. Features like multiple-pipeline execution, large L2 Cache capabilities greatly improve cluster performance. On the primary memory part , more the memory lesser the HDD accesses and in turn faster execution. One peculiar fact is that joining 100 486 machines wont give performance greater then a Pentium II processor running at 400 MHz. So CPU selection has to be done intelligently.

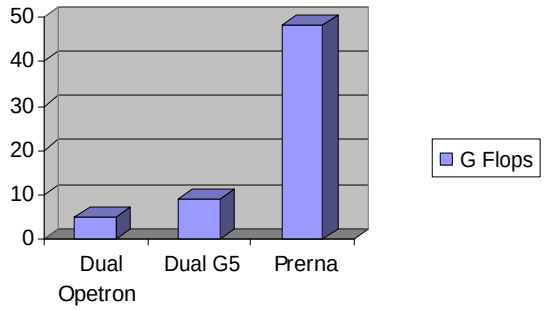
Implementation

Once the analysis part is over, then joining the systems and installing the required is fairly easy part. Now comes the challenging part i.e. testing the system. There are many benchmarks available such as Linpack, openmosix stress test etc. One very efficient for testing is running the application on one node and then executing it on the cluster. The gap in time of execution on the two can well define the performance. If the performance in the cluster is not up to the mark then the analysis part point three can be worked upon, even tuning of the software can result into better result.

Case Study : Prerna – The DAVIET cluster

Prerna is a 23 node Linux based cluster at DAV Institute of Engineering & Technology. The hardware configuration of each node is: Intel Pentium4 @ 1.80GHz, a total of 5.9 GB DDR RAM and about 0.9TB secondary storage with a 100 Mbps Ethernet network. The prime area of application is Fiber-optic simulations and IT research. On average the performance achieved is 40+ G Flops.

The first step followed was to collect the information on various technologies available. PVM was mixed with windows and MPI with Fedora, GridIron and Mandrake CLIC were also tested and then Knoppix with openMosix. The later combination forms the heart of the cluster at present. The hardware configuration i.e. 1.8 GHZ processors were ought sufficient for the processes. The current version Prerna V.1. Sits on top of kernel 2.2.21. Things are really working fast for it and it if finding it's way soon in to its next version Prerna V.2. which supports MPI and higher number of nodes. The configuration is set to be revised and will rock to 3.0 GHz HT processors soon.



System	Performance	Price
Prerna	48 G Flops	~Rs. 250,000
Dual processor G5 Apple xServe	9 G Flops	~Rs. 188,000
Dual 64-bit Opteron Server	5 G Flops	~Rs.3,75,000

Source DIGIT magazine September 2004
Note: The prices quoted here are subjected to change. These are based around the values when the paper was being writeen.

Test	One Node	Prerna Cluster
SETI@Home(50 data Units)	~275 Hours	17 Hours 45 Minutes(25 node)
C Code Test	1 Hour 10 Minutes	10 Minutes 02 Seconds (with just 8 nodes)

Advantages :

There are endless benefits of a computer cluster. The first and foremost point is that the exposure which the students get in parallel computing without the need of having to work on a huge monolithic machine. And this is also gives a huge boost to the research facilities. The future sees clusters replacing the mighty servers, clients will avail the super-number crunching power without have to process data at it's on. A lot of research is being done on Load Balancing on servers these days and clusters are the right tool for testing such stuff. Further research in Gnome and D.N.A. simulations are a very promising applications of clusters.

One more future prospect of a cluster is that it can be mixed with a SIMPUTER and the later can draw the computation power from it. This will greatly reduce the price of SIMPUTER and thus the real aim of this device can be achieved. And not to forget the arena of nuclear bomb simulations, medical drug simulations and aerodynamics simulations. The list is just endless.

At present clusters just seem to be promising and even deeper.

Bibliography

1) The Beowulf group supported by SYLD www.beowulf.org

2) The official MPI site - www.mpi-forum.org

3) The PVM site -netlib.org/pvm3
4) The openMosix official site www.openmosix.sourceforge.org
5) www.bofh.be/clusterknoppix
6) www.rocksclusters.org
7) The DIGIT computer magazine www.thinkdigit.com
8)Parallel programming - mhpcc.edu/training/workshop/parallel_intro/MAIN.html
9) The PC Quest magazinewww.pcquest.com

Other clustering details - uwo.ca/wnews/issues/2001/apr19/centre/index.htm

The house of The Cluster Prerna - www.davietjal.org/prena.html
Operating System Concepts by :Syberghautz