Parallel Panther Beowulf Cluster

Dheeraj Bhardwaj <dheerajb@eiu.edu> Version 1.1, 4 September 2000

This document is about building a NO cost scalable parallel computer. It describes the details and art in making Parallel Panther at Eastern Illinois University.

1 Introduction

What is Parallel Panther

Parallel Panther is a NO cost Beowulf class supercomputer. Beowulf is a multi computer architecture, which can be used for parallel computations. It is a system built using commodity hardware components such as PC's connected via Ethernet or some other network. The hardware used in Parallel Panther is the discarded hardware by various labs and faculty members.

Hardware used to build Parallel Panther

8 Gateway2000 PCs from Gateway <http://www.gateway.com>, each consists of:

1. 133 MHz Intel Pentium Processor
2. 64 MB RAM
3. 3GB IDE Hard Disk
4. 3COM 3c509c-TX Fast Ethernet adapter (10/100 Mbits/sec)
5. 24x IDE CDROM drive
6. 1.44 MB floppy drive
7. Cables

In addition, the following are also used:

1. One 24-port 100Mbps Ethernet SuperStackII 3300 XM switch
2. One monitor, keyboard, mouse, extra hard disk of 8.4 GB and 128 MB of RAM for one PC which is used as one of the servers.

Operating System on Parallel Panther

Parallel Panther is powered by **GNU-Linux!** In specific**,** RedHat 6.2 distribution <http://www.redhat.com>

GNU-Linux is a free version of the Unix operating system which has Linux kernel written by Linus Tovalds <torvalds@transmeta.com> and other volunteers. Most of the programs running under GNU-Linux are generic Unix freeware, many if them from the GNU project <http://www.gnu.org>.

Software Installed on Parallel Panther

1. Base package from RedHat Linux 6.2 distribution.
2. Freeware GNU C/C++ compilers.
3. F77 to C converter
4. Freeware Message Passing Interface (MPI) library for parallel programming in C/C++/Fortran77/Fortran90 <http://www.mcs.anl.org>.
5. Freeware Parallel Virtual Machine (PVM3) library for parallel programming in C/Fortran <http://www.netlib.org>.
6. Scientific Visualization Software

Cost of Parallel Panther

Approximately NO cost.

2. Art in Making Parallel Panther

2.1 Setting up the Hardware

Here are the main steps in setting up the hardware:

1. Moved the machines from various labs in the campus to the parallel processing lab, mounted them on the rack, names and numbered them.
2. Setup the 24-port network switch and connected one of the port to the main box by a cross over cable. Other 8-ports were connected to the Ethernet adapters of the machines
3. A single monitor, keyboard and mouse were connected to one of the machines which has 128 MB RAM, 8.4 GB extra hard disk and a high end graphics card.
4. Powered all the components.

2.2 Setting up the Software

Here are the steps, which made the reality come true:

1. OS installation: RedHat Linux 6.2 was installed on all the machines by connecting all the peripherals such as monitor, mouse and keyboard. Most of the hardware was automatically detected, so main focus was on partitioning the drive and choosing the relevant packages to be installed. Its is very important to choose partition size which are correct for the need because it might be very difficult to change this at a later stage when the cluster will be in the functional mode. Following is the list of partitions:

   - / -500MB. This / partition contains /bin, /boot/, /dev., /lib, /root, /sbin, /var and /home directories with their contents.

- /usr – 1.5 GB. This /usr partition were created by keeping in mind that most additional rpm's will install in /usr.

- swap – Swaping is really bad for the performance of the system. Unfortunately there might be a time when the machine is computing a very large job and just don't have enough memory. Since the machines have 64 MB RAM, it was realized that a 128MB of swap partition was a good idea.

- /export/home - Rest part of the disk. This partition was used as users home on individual machines. If needed it can be NFS mounted for additional user space. One of the machines had 8.4GB of addition hard drive, that was used for common home area for users.

2. Network Configuration: During OS installation IP addresses and nodes names were assigned. Following are the nodes names of Parallel Panther

Cheeta	(139.67.32.150)	Cat	(139.67.32.154)
Tiger	(139.67.32.151)	Jaguar	(139.67.32.155)
Lion	(139.67.32.152)	Kitten	(139.67.32.156)
Leopard	(139.67.32.153	Cub	(139.67.32.157)

For the configuration, the following files were modified: /etc/sysconfig/network and /etc/sysconfig/network-scripts/ifcfg-eth0. Here are these two files /etc/sysconfig/network:

```
NETWORKING=yes
HOSTNAME=cheeta
GATEWAY=139.67.32.1
NISDOMAIN=parallel
FORWARD_IPV4=yes
```

/etc/sysconfig/network-scripts/ifcfg-eth0

```
DEVICE=eth0
BOOTPROTO=static
BROADCAST=139.67.32.255
IPADDR=139.67.32.150
NETMASK=255.255.248.0
NETWORK=139.67.32.0
ONBOOT=yes
```

/etc/hosts.equiv: In order to allow remote shells (rsh) from any node to any other in the cluster, for all users, we should relax the security, and list all hosts in /etc/hosts.equiv.

3. NFS configuration: We have used fully local OS install configuration for this beowulf cluster In this setup all the participating machines have their own disks with operating system and swap locally and only /home and /usr/local off the

servers. We use **N**etwork **F**ile **S**ystem (NFS) for mounting /home and /usr/local partitions. The advantage of this setup is no NFS traffic and the disadvantage of complicated installation and maintenance is over taken by writing shell scripts such as rcmd (remote command) and rsync, which could update all file systems. Following were the steps used to configure NFS:

We selected *cheeta* as server node and exported the /home1 partition on additional 8.4GB disk and /usr/local partition by modifying the /etc/exports as:

/etc/exports looks like:

```
/mnt/cdrom   (ro)
/home1              (rw,no_root_squash)
/usr/local          (ro,no_root_squash)
```

We use automounter (*autofs*) for mounting the exported /home1 partition on the clients. Fist we installed rpm of autofs from the RedHat Linux 6.2 CD. *autofs* automatically mounts the various user partitions on demand. Automounter gives better NFS performance. For configuring *autofs* , modified /etc/auto.master and /etc/auto.home files by additing following lines:

/etc/auto.master:

```
/home  /etc/auto.home
```

/etc/auto.home:

```
/home cheeta:/home1
```

For mounting /usr/local file system on clients we used traditional NFS mount, i.e. by appending /etc/fstab files with the following additional line:

```
cheeta:/usr/local          /usr/local     nfs     defaults       1 2
```

4. NIS configuration: The Network Information Service (NIS) is an administrative database that provides central control and automatic dissemination of important administrative files. NIS converts several standard UNIX files into databases that can be queried over the network. The databases are called NIS maps. Following are the main steps for configuring NIS:

Configuration of NIS master server:

We selected NIS doamin name “parallel” and created a file /etc/defualtdomain and put the doaminname entry as parallel. Then we set the domainname as:

```
# domainname parallel
```

We created a directory /var/yp/src and copied following files in it from /etc directory

```
#cp /etc/passwd              /var/yp/erc/.
#cp /etc/shadow              /var/yp/erc/.
#cp /etc/ether               /var/yp/erc/.
#cp /etc/hosts               /var/yp/erc/.
#cp /etc/group               /var/yp/erc/.
#cp /etc/network             /var/yp/erc/.
#cp /etc/protocols           /var/yp/erc/.
#cp /etc/rpc                 /var/yp/erc/.
#cp /etc/services            /var/yp/erc/.
#cp /etc/auto.master         /var/yp/erc/.
#cp /etc/auto.home           /var/yp/erc/.
```

We made the following changes in the /var/yp/Makefile

```
YPSRCDIR    = /var/yp/src
YPPWDDIR    =/var/yp/src
```

Then we initiated the yp services as

```
# /usr/sbin/ypinit –m
```

It asked for the NIS server name. We named the NIS server as cheeta and followed the following steps:

```
#cd /var/yp
#make
```

This created the database for the NIS services. Then we started the NIS services from /etc/rc.d/init.d.

On the clients we set the defaultdomain name as parallel and started NIS services from system setup by following command

```
#setup
```

After starting the NIS services. To bind the clients with NIS server we start ypbind as

```
#/etc/rc.d/init.d/ypbind start
```

These steps configured the NIS. To check the NIS services, try these

```
#ypcat passwd
#ypmatch username passwd
```

5. BIOS configuration: For booting machines without monitor, keyword and mouse, BIOS was configured on all the machines. We connected the monitor, mouse and keyboard to the nodes and configured the BIOS for no halt in the absence of keyboard, mouse and monitor.

6. *MPI* Installation: We downloaded the freely available version of MPI (Massage Passing Interface) MPICH-1.2.0 from http://www-unix.mcs.anl.gov/mpi/mpich Gunzip, untar, configure and make were the steps used for the installation of mpich-1.2.0 in /usr/local/mpich-1.2.0.

7. *PVM* installation: Parallel Vitual Machine (PVM) installation was automatic. During the selection of packages, we selected clustering tools and it installed in /usr/share/pvm3. Later we moved pvm3 directory to /usr/local/pvm3.

8. *XPVM* installation: X-windows version of PVM also installed automatically like PVM

9. *http* server configuration: In order to configure http server on server node i.e. cheeta we modified /etc/httpd/conf/http.conf file with server name as cheeta. We have all the information available about the machine, its usage and tutorials on how to write programs using MPI and PVM.

10. Scripts for single system image: We have written few scripts such as rcmd, rsync etc for monitoring the machine from console on command line.

11. Condor installation: Condor is planned to install as a resource management tool.

12. bWatch: bWatch is a GUI Beowulf Cluster Monitor. It displays load averages, memory swap, and number of processes and users for all nodes in a single window.

3. Benchmarks

The peak performance of Parallel Panther is 2.1GF. Parallel Panther is now fully functional and ready for the benchmarking. Standard benchmark suits need to be run for finding the sustained performance.

4. Usage of Parallel Panther

Parallel Panther will be used mainly for teaching and research purposes. Its will be used for development of parallel scientific applications for small sizes. Which can later be benchmarked on the supercomputers available with central facilities such as NCSA at University of Illinois at Urbana Champaign. The parallel codes written in MPI and PVM can directly be ported on any parallel computer, which supports MPI and PVM irrespective of OS, hardware, system software etc.

5. Futures Plans

We have planned to expand the size of the machine by adding more nodes as and when they will be available from various labs and faculties in campus. This architecture of the machine provides a flexibility of adding more nodes to it with less efforts.

The machine and documents are developed and maintained by:

Dr. Dheeraj Bhardwaj
Information Technology Services
Eastern Illinois University
Charleston, IL 61920
e-mail: dheerajb@eiu.edu
http://www.ux1.eiu.edu/~csdb2