;login:

THE MAGAZINE OF USENIX & SAGE

August 2001 • Volume 26 • Number 5

Special Focus Issue:Clustering

Guest Editor: Joseph L. Kaiser

inside:

USENIX & SAGE

The Advanced Computing Systems Association &
The System Administrators Guild

the PIRUN cluster at Kasetsart University: the most powerful supercomputer in Thailand

by Putchong Uthayopas

pu@ku.ac.th

Putchong Uthayopas has been involved in Beowulf clustering technology for the past 5 years. He is the head of the SCE (Scalable Cluster Environment) project. Currently, he is an assistant professor at the Department of Computer Engineering, Faculty of Engineering, Kasetsart University, Thailand.

and Somsak Sriprayoonsakul

Somsak Sriprayoonsakul is in the Master Degree Program ofthe Department of Computer Engineering at Kasetsart University.He has worked as a system administrator for numerous Linux Servers and has helped build many clusters. He is the author of a scheduling system on Beowulf Cluster called SQMS (Simple Queueing Management System) which is a part of the SCE integrate software environment for Beowulf Clusters.

In the past, high-performance computing was considered by many as too costly to be widely practiced in Thailand. Most of the high-performance computing research in Thailand has been done by a small group of people who used small commercial supercomputers operated by government agencies and universities. These machines had only about four to six processors per machine with a computing speed aggregate of only about 2–3Gflops. As Beowulf clusters become widely used by organizations around the world, the cost problem mentioned earlier is resolving itself. Due to the high performance and low cost of these types of systems, Beowulf technology seems to be the solution for developing countries like Thailand.

Beowulf Cluster Development at Kasetsart University

Kasetsart University is the second oldest university in Thailand, established on February 2, 1943. Current enrollment is around 25,000. Our research laboratory (the Parallel Research Group) was founded in 1996 to explore Beowulf clustering technology and its possible applications. We have since built many clusters.

We started our cluster adventure by building a small five-node 486SX Beowulf cluster that ran MPICH. The first machine's performance was around 2–3Mflops. It was nevertheless a real thrill to finally have a machine for which we could write a parallel program and then actually run it. The next several machines that we built grew rapidly in terms of size, complexity, and performance. Finally, the moment that we had long been waiting for arrived when, in 1999, the university's computing center had a budget to build a 72-node Beowulf cluster. With this size, the machine is definitely the most powerful supercomputer in Thailand.

The PIRUN Beowulf Cluster

One of the most important tasks was to find a good name for the machine. After several serious discussions among the group, we decided to call it PIRUN (Pile of Inexpensive and Redundant Universal Nodes) Beowulf cluster. Coincidentally, PIRUN is also the name of the Thai god of rain and the god image that our university uses in all of its logos.

The goals in building this machine were to:

- Build a system to serve as a centralized Internet super-server for 15,000–20,000 Internet users at Kasetsart University (KU).

- Provide a world-class supercomputing facility for researchers at KU in areas such as computational chemistry, computational fluid dynamics, bioinformatics, and computer science.
- Build a large PC cluster to be used as a test-bed for cluster computing technology.

PIRUN Beowulf Cluster Hardware

The PIRUN cluster system comprises:

- Seventy-two compute nodes built with Pentium III 500MHz, 128MB RAM per node, ASUStek P2B series motherboard, that support wake on LAN and hardware monitoring chips.
- Three file server nodes using a dual Pentium Xeon 500MHz server with hardware RAID. Each server node has about six Ultra SCSI disks of 9GB for a total of about 54GB per node.
- KVM (Keyboard/Video/Mouse) switch with daisy-chained capability. A total of 10 KVMs have been chained to centralize the console access to a single node.
- Four 24-port, stackable, Fast 3Com Superstack II Ethernet switches are used to link the system together.

We decided to reduce some cost by having this system consist mostly of diskless nodes. We developed an in-house tool to aid in installing the system ourselves. These tools and much more are available for download at the main Web site: *http://smile.cpe.ku.ac.th*. The whole configuration cost around $200,000 in 1999.

We decided to reduce some cost by having this system consist mostly of diskless nodes.

Planning the System Configuration

We decided to divide the 72 nodes into three sets of 24 nodes each. Each node is named CPIRUNxx, where xx is the corresponding IP number of that node. For example, CPIRUN11 has IP number 192.168.10.11. (IP addresses have been changed to protect the innocent.) The file servers are named FPIRUN1, FPIRUN2, and FPIRUN3. Each set of 24 nodes uses the diskless file system (/, /usr, /lib, /var, etc...) from each FPIRUN to lessen the file server load. The organization is:

- CPIRUN11–CPIRUN34 use file on FPIRUN1
- CPIRUN41–CPIRUN64 use file on FPIRUN2
- CPIRUN71–CPIRUN97 use file on FPIRUN3

The tool we used to build the diskless file systems is called the "Diskless Cluster Suite" (version 1.2) and was built by us for this purpose. This tool is also available for download from our site and from the Tucows site (*http://www.tucows.com* and search for the name under the Linux category).

For our setup, FPIRUN1 also provides the common space for the mail spool directory /var/spool/mail for all nodes. All 72 nodes mount the users' working directory (/home) from each FPIRUN. However, there are special partitions for many particular purposes. For example: FPIRUN1 exports /home directory for staff members. FPIRUN2 holds /home2, which contains temporary space provided for research projects that use a very large disk space. Finally, FPIRUN3 holds /home3 for student users. (This is not guaranteed to be as reliable as the staff volume.)

Building and Installing the System

Around December 1999, the equipment finally arrived. The first task was to connect all the hardware together, so we rounded up a group of volunteer students and faculty members and had fun getting things out of boxes, making cable connections, and putting lots of systems on racks. Things went very smoothly, and we were able to finish the installation within a few days. The first problem we encountered was the power system; there were some mistakes in balancing the power phase, and things needed to be rearranged. So, first lesson: be careful, these clusters really need power and a large floor space.

Be careful, these clusters really need power and a large floor space.

The software installation was our next consideration. First, the three file servers FPIRUN1, FPIRUN2, and FPIRUN3 were installed using Linux. We chose Linux RedHat 6.2 (the best and newest at that time). Second, the Diskless Cluster Suite tool was used to generate the initial configuration. Due to the complexity of installation, some configuration had to be performed manually to meet our requirements. For convenience, we allowed each FPIRUN to rsh to each other without a password prompt by creating the file /root/.rhosts that contained the fpirun1, fpirun2, fpirun3 hostnames. Of course we also added fpirun1, fpirun2, fpirun3 hostnames to /etc/hosts. For better security, this has now been replaced by SSH.

To create the diskless configuration on the nodes, we used Diskless Cluster Suite three times on each of the file servers, each time creating a configuration of 24 nodes. All we needed to do was to merge all configurations into one and distribute it to each diskless space (each directory under /tftpboot of each FPIRUN). For the DHCP configuration, the Diskless Cluster Suite software already did this part for us. We only needed to boot each diskless sequentially one-by-one the first time so that the DHCP server could assign a correct IP for each node. DHCP always remembers the client's MAC address, so next time, all the nodes could be booted at once. (We set the DHCP lease time to infinite to get this effect.)

For the hosts' configuration, we had to fix the /etc/hosts to contains all hostnames and IPs of all nodes. The Diskless Cluster Suite had already done this for each node set. Hence, the only task we needed to do manually was to merge the /etc/hosts from all hosts into one and distribute it back to each FPIRUN file server.

File System Configuration

Diskless Cluster Suite generated all our file system configuration needs for each 24-node cluster as well. First, we needed to fix /etc/exports for each FPIRUN to allow mounting of the /home, /home2, /home3 partitions by adding these lines to /etc/exports on FPIRUN1:

```
/home 192.168.3.41(rw,no_root_squash)
/home 192.168.3.42(rw,no_root_squash)
/home 192.168.3.43(rw,no_root_squash)
 .....
/home 192.168.3.71(rw,no_root_squash)
 /home192.168.3.71(rw,no_root_squash)
.....
```

We did the same thing for FPIRUN2, FPIRUN3 but changed from /home to /home2 and /home3, respectively. This can be done easily with regular UNIX commands. On FPIRUN1 we typed:

```
% rsh fpirun2 cat /etc/exports | grep home | \
      sed "s/home2/home/g" >> /etc/exports
% rsh fpirun3 cat /etc/exports | grep home | \
      sed "s/home3/home/g" >> /etc/exports
```

This command uses cat to print out all contents of /etc/exports in FPIRUN2, then we grep only the home2 entry, and change it to home. The same goes for FPIRUN3, but home3 changes to home. Of course, we need to do this on FPIRUN2 and FPIRUN3 as well. In total, we needed to issue these commands six times.

Next, we modified the /etc/fstab file on each diskless node to mount /home, /home2, and /home3 from each FPIRUN properly. This was also done by adding the entry (CPIRUN11 /etc/fstab (192.168.3.10, 192.168.3.40, 192.168.3.70 are FPIRUN1, FPIRUN2, FPIRUN3, respectively)) as follows:

```
192.168.3.10:/dev/nfsroot     /              nfs      defaults     0 0
none                          /proc          proc     defaults     0 0
none                          /dev/pts       devpts   defaults     0 0
192.168.3.10:/tftpboot/usr    /exportusr     nfs      defaults     0 0
192.168.3.10:/usr/local       /usr/local     nfs      defaults     0 0
192.168.3.10:/usr/software    /usr/software  nfs      defaults     0 0
192.168.3.10:/home            /home          nfs      defaults     0 0
192.168.3.40:/home2           /home2         nfs      defaults     0 0
192.168.3.70:/home3           /home3         nfs      defaults     0 0
192.168.3.10:/var/spool/mail  /var/spool/mail nfs     defaults     0 0
```

The last four entries were added by us, and the first six lines were created automatically by the Diskless Cluster Suite. The same goes in CPIRUN41, and CPIRUN71.

In order to be able to run the MPICH program, all diskless nodes must be able to rsh to each other without a password prompt. The Diskless Cluster Suite already does this for each 24-node set; our task was again to merge them into one. This can be done easily by issuing the command:

```
% rsh fpirun2 cat /etc/hosts.equiv >> /etc/hosts.equiv
% rsh fpirun3 cat /etc/hosts.equiv >> /etc/hosts.equiv
```

Then, we needed to delete the fpirun1, fpirun2, and fpirun3 entries from /etc/hosts.equiv since we prohibit user access to FPIRUN. After finishing all that, we could easily copy all these files back in place.

```
% rcp /etc/hosts.equiv fpirun2:/etc/hosts.equiv
% rcp /etc/hosts.equiv fpirun3:/etc/hosts.equiv
```

Booting the Nodes

We used the Diskless Cluster Suite to create a boot disk for each node. This was time-consuming since we had to create 72 disks for 72 nodes. This process can be sped up using a smart parallel technique. First, create the first 24-disk set. Then, dump the disk image to a file using the following command:

```
% dd if=/dev/fd0 of=/home/diskless_image
```

Boot the first 24 nodes. Then, insert a new set of disks to these 24 nodes and have all nodes dump the disk image to the disk using the same command.

```
% dd if=/home/diskless_image of=/dev/fd0
```

These new disks, created simultaneously, are produced 24 times faster.

User Account Synchronization

We synchronized the user accounts by distributing /etc/passwd, /etc/shadow, and /etc/group to each diskless space. Using this replication instead of an NIS-based system will generate less traffic and will be faster since all user logging and verification is done on the local node. We centralized the account information at FPIRUN1, so any changes must be done there first. The later file distribution is done using rcp between file server nodes and plain cp in the diskless configuration under /tftpboot, using an hourly crontab to copy each file to each diskless space. For example, at FPIRUN1 we add this script:

```
#!/bin/bash
#/etc/cron.hourly/copypassword
for i in /tftpboot/*.*.*.*; do
      cp -f /etc/passwd /etc/group /etc/shadow $i/etc/
done
```

At FPIRUN2 and FPIRUN3, we needed to add rcp to copy the files from FPIRUN1 to their own space.

```
#!/bin/bash
#FPIRUN2 and FPIRUN3 /etc/cron.hourly/copypassword
rcp fpirun1:/etc/passwd fpirun1:/etc/shadow fpirun1:/etc/group /etc/
for i in /tftpboot/*.*.*.*; do
      cp -f /etc/passwd /etc/group /etc/shadow $i/etc/
done
```

For configuration files like /etc/bashrc, /etc/profile, and /etc/csh.cshrc we created a /tftpboot/config directory in each FPIRUN to hold these files and used hard links from each of these files in diskless space to the file in /tftpboot/config. This will ease configuration updates by changing the file in /tftpboot/config and then distributing the file to all diskless nodes. We cannot use a soft link since the directory structure on the diskless nodes and servers are different.

Changing Passwords

Users can change passwords at FPIRUN1 only; /etc/passwd will be updated every hour. There is a catch-22, however: we prohibit user access to FPIRUN1. This can be done by creating /etc/nologin on FPIRUN2 and FPIRUN3. FPIRUN1 already has /etc/profile to deny user login. So we modified /etc/profile so that if a user logs in, it will automatically run the command passwd to change the password. The script is below and can be handy.

```
if [ "$USER" != 'root' ];then
      /usr/bin/passwd
      /usr/bin/logout
fi
```

We prohibit all access via Telnet, rlogin, and rsh from outside the cluster by using tcp-wrappers (/etc/hosts.allow, /etc/hosts.deny). We use an SSH 2.4.0 server for remote login.

Finally, we use DNS round-robin to distribute the hostname to the client. All CPIRUN nodes will have the name cpirun.ku.ac.th associated with all CPIRUN IPs. Below is the output from the nslookup command:

```
Name: cpirun.ku.ac.th
Addresses: 192.168.3.47, 192.168.3.48, 192.168.3.49, 192.168.3.50,
192.168.3.51, 192.168.3.11, 192.168.3.12, 192.168.3.13, 192.168.3.14,
192.168.3.15, 192.168.3.16, 192.168.3.17,  192.168.3.18, 192.168.3.19,
192.168.3.20, 192.168.3.21, 192.168.3.22, 192.168.3.23, 192.168.3.24,
192.168.3.25, 192.168.3.26, 192.168.3.27,  192.168.3.28, 192.168.3.29,
192.168.3.30, 192.168.3.31, 192.168.3.32, 192.168.3.33, 192.168.3.34,
192.168.3.41, 192.168.3.42, 192.168.3.43, 192.168.3.44, 192.168.3.45,
192.168.3.46
```

At this point, the machine is running, so it's time to do something useful.

How We Use the PIRUN Beowulf Cluster

One of the problems that formerly prevented us from offering any course in parallel and distributed computing was the lack of a real parallel computing platform. In Thailand, we expect that manpower need will be greatest in the areas of parallel scientific computing and system engineering. Therefore, the courses emphasize parallel programming and algorithms more than parallel architecture. Also, the interest in cluster computing and the building of cluster infrastructure has created many new kinds of research programs and new areas of research, some of which are listed below:

- Parallel software tools and environment, cluster administrator tools, cluster integration tools, cluster middleware
- Parallel and distributed applications
- Internet search engine, parallel text search engine, Web infrastructure
- Pollution modeling, fluidized bed simulation in chemical engineering, molecular dynamics simulation, computational fluid dynamics application in vehicle design, and heat analysis in electronics industry

Conclusion

We have gained invaluable experience from the Beowulf cluster system at KU and have learned that careful planning is essential in setting up a large Beowulf system. If everything has been carefully planned, this kind of system can be put in place within one or two days.

Because of our expertise in cluster software tools, initial software installation was done quickly. Without this kind of automated scripting, the installation would be very tedious and long. However, since there is always something misconfigured or totally missing, it takes awhile for the system to become smoothly operational for users. User feedback is also important to improve the operation of the system.

Overall, we have had a very positive experience; the Beowulf cluster system has brought many benefits to our organization in terms of cost reduction, producing new expertise, and creating new projects. It was almost impossible a few years ago for our university to possess the most powerful supercomputing system in the country. Moreover, it was hard to imagine that building such a machine could be done easily using just a bunch of PCs on rack. Linux and the free software movement have actually created a miracle here. We expect use of this class of system in Thailand to increase in the coming years. There is currently an effort to form a communication network among researchers in Thailand. More information about cluster computing activities in Thailand can be found at *http://tfcc.cpe.ku.ac.th*.

Acknowledgments

The PIRUN system has been maintained by many students and staff members. First, we would like to acknowledge the contribution of Professor Yuen Povarawan, director of the KU computing center who helped push until this project became a reality; my friend, Assistant Professor Surasak Sa-nguanpong, the co-project technical leader, who spent a lot of his time selecting the most effective hardware combination for the project; Thara Angskun and Jakchai Sonakul, who first installed the system; Sugree Phatanapherom and Theevara Vorakosit, who helped tweak the machine and catch some bugs. Also, we appreciate the many, many students who helped set up the machine in the first phase. Without their kind help, the setup would have been hard and painful. Finally, we would like to thank Dr. Thomas Sterling, father of the Beowulf cluster, who inspired us all in doing this. In fact, he kindly visited us in Thailand and gave an inspiring talk at Kasetsart University. He also visited the PIRUN cluster site when we had our first 16 nodes set up and made many useful comments about the future of this project.

REFERENCES

T. Sterling, D. Becker, D. Savarese, J. Dorband, U.A. Ranawake, and C.E. Packer, “Beowulf: A Parallel Workstation for Scientific Computation,” in *Proceedings of the International Conference on Parallel Processing*, 1995.

Barry Wilkinson and Michael Allen, *Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers* (Prentice Hall, 1999).

William Gropp, Ewing Lusk, and Anthony Skjellum, *Using MPI: Portable Parallel Programming with the Message-Passing Interface* (Cambridge, MA: MIT Press, 1999).

Putchong Uthayopas, Surasak Sanguanpong, and Yuen Poovaravan, “Building a Large-Scalable Internet Super Server for Academic Services with Linux Clusters Technology,” in *Proceedings of Internet Workshop2000*, Tsukuba, Japan, February 15–17, 2000.