Heterogeneous Clustering - Operational and User Impacts

Sarita Salm
Sterling Software
MS 258-6
Moffett Field, CA 94035-1000
sarita@nas.nasa.gov
http://science.nas.nasa.gov/~sarita

ABSTRACT

Heterogeneous clustering can improve overall utilization of multiple hosts and can provide better turnaround to users by balancing workloads across hosts. Building a cluster requires both operational changes and revisions in user scripts.

Introduction

The work presented in this paper was done at the Numerical Aerospace Simulation Facility (NAS). The goal of the facility is to conduct pathfinding research in large-scale computing. The facility serves customers in government, industry, and academia with supercomputer software tools, high-performance hardware platforms, and 24-hour consulting support to address the ever-increasing complexity of aerospace design problems.[1]

The NAS cluster was designed to minimize the average time it takes to complete a job, while providing compute cycles for all jobs. Minimizing idle time and the impact of system failure are important steps toward this goal.

When planning the heterogeneous cluster, impact from failure of a single host needed to be considered. Additionally, when shutting down a single host, work needed to be preserved. User impacts needed to be minimized. We faced issues with simplifying file access, user account consistency, and heterogeneity of systems. Added attributes were designed to provide flexibility for the user to continue work as they previously had or to create cluster aware scripts.

Environment

The NAS cluster consists of four Cray Y-MP J90 systems, two Cray Y-MP C90 systems, and two SGI Origin 200 systems (See Figure 1). The J90 mainframes and the C90 mainframes serve as hosts for user jobs and file systems. The large C90 system runs UNICOS 10.0.0.3, while all other Cray hosts run UNICOS 10.0.0.2. The SGI Origin 200 systems remain at IRIX 6.4 and SGI Failsafe 1.2.

The SGI Origin 200 systems utilize SGI Failsafe software. SGI Failsafe software allows two different machines to assume a single IP address, one at a time, in order to provide a high-availability system. This system provides access to a central information repository used to make desired changes to user udb entries. It also provides access to Portable Batch System (PBS) 1.1.10 server software, pbs_server. PBS software provides the framework needed to track resources and schedule batch jobs.

The SGI Origin 200 system, which is reached through the logical IP address, runs the PBS 1.1.10 server, pbs_server, and a job router. All

[1] adapted from www.nas.nasa.gov web page.

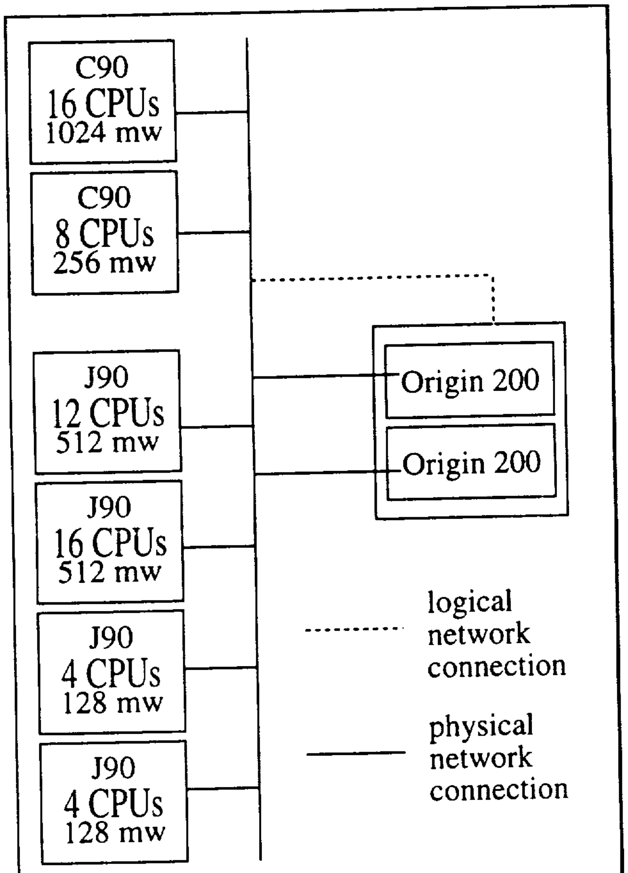


Figure 1

Cray systems run the PBS 1.1.10 client, pbs_mom, and a job scheduler. The job scheduler chooses which jobs will start. Pbs_mom starts and tracks jobs. Pbs_server provides a database to keep track of jobs. The job router verifies that each new job can run on one or more systems within the cluster, deleting those jobs that cannot run on any system.

Benefits

Effective scheduling of jobs and resources within the cluster balances the job load across multiple hosts. This minimizes the chance that hosts will sit idle while work remains queued on other systems, and results in better resource utilization.

The NAS cluster's job scheduling software provides flexibility in configuring job choice on a host-by-host basis. Under this framework, smaller jobs may run on smaller systems, while larger jobs can obtain more of the resources of the larger systems. When configured appropriately, the scheduler can also prevent a job from using all of one system's memory while using a fraction of the CPUs. This helps ensure minimal idle time on systems, more efficient use of the systems, and a reduction in average job turnaround time.

Previously, users would look at the load across the systems to try to determine where their job would run first. This was an easy task when systems were obviously unbalanced, but was much more difficult when all systems were overloaded. Job scheduling allows a job to run, in turn, on the first system that can accommodate the job's needs.

Clustering also allows work to continue on systems while another system is not available. It does this with no intervention needed from users who run cluster aware jobs.

Reliability Improvements

The four J90s used in this heterogeneous cluster were previously run as a J90 cluster. In this homogeneous cluster, we also used PBS. Home file systems are mounted via NFS from one host to the other three hosts.

Within the J90 cluster, failure of the host which ran the PBS server caused a number of problems. One such problem occurred when a job finished and the system running the PBS server was unavailable. The PBS server did not receive job termination notification, and the job would be rerun from the beginning.

To minimize this problem, we added two SGI Origin 200 systems and the SGI Failsafe product to the heterogeneous cluster. I modified PBS software to work with SGI Failsafe constructs and made changes to allow information to return to the logical IP address, even when the original point of contact was a physical IP address. Lastly, I wrote scripts to perform needed status checks and to perform the fail over and fail back functions.

For the most part, this solution has worked well. A notable exception occurs when a system that is running jobs started by PBS crashes. This situation causes the PBS server to become unresponsive. We run a manual workaround each time this situation occurs. This workaround modifies the PBS server database to requeue jobs that are listed as running on the failing host. The server is then killed, in a way that does not rewrite the database, and immediately restarted to read in the modified database.

Single Client Shutdown/Startup

PBS was designed to work with systems in which all clients go down at the same time. On hosts running UNICOS, all jobs are checkpointed before a full shutdown. Work was lost because PBS does not have a similar mechanism when shutting down a single client.

A script is run to checkpoint jobs prior to shut down. This script obtains the list of jobs started by PBS on the host and then uses PBS's **qhold** to checkpoint each job. During startup, PBS's **qrls** is used to release each of these previously held jobs while not releasing any other jobs.

File systems

Users of the NAS cluster currently have three home file systems, one for each of the three hosts that allow interactive logins. All J90 systems use home file systems (locally, and via NFS) located on the 12-CPU J90 system. Loss of the 12-CPU J90 system causes problems for all of the J90 systems.

When users first used the cluster, they expected jobs submitted on the large C90 to run on the same system. They became confused when their directories from the large C90 were not found. They did not notice the banner which indicated that their job had run on one of the other hosts. Links, */R/hostname/u/username*, were setup so that users could consistently access their home file systems from each host, whether via NFS or locally.

Users are provided with a significant amount of temporary local disk space. We recommend using this local disk for many reasons including improving the performance of jobs. Additionally, on large files, use of **rcp** is faster than NFS. Lastly, relying on NFS for consistent access to data can cause a job to fail if the host which provides that data is not accessible, either through failure of the network or failure of the host.

To resolve availability issues and to reduce user confusion, we have discussed the possibility of creating a single home file system for each user on an SGI Failsafe system. At the CUG Origin 2000 workshop (October 1998), **cxfs** was introduced. This product, which I understand will have the functionality of NFS and performance close to that of local disk, could also resolve file system issues (though it may be an IRIX specific product).

Security

The PBS software relies on the ability to **rcp** files between the host running the PBS server and the host running the PBS client software. It also needs this ability between the host running the PBS server and hosts from which users submit jobs. In the NAS environment, it is not feasible to expect users to add all hosts to their **.rhosts** file. To accomplish this with little impact to the user, all hosts within the cluster have been configured to trust one another.

Accounts/Accounting

Making user accounts consistent has been a major task. At first, each system had its own list of users. We did not want to lose track of users by regenerating the same udb on different systems, as there were things such as mail, crontabs, and file systems that needed to be cleaned up. Additionally, we cannot use the same udb information on all systems, as some systems allowed interactive access while others did not and interactive limits on the J90 systems differed from those on the C90 systems.

We needed a way to update a single user across all systems that would be minimally impacted by system availability. We already used Failsafe to support the batch server, so we added a file system to that system to be NFS-mounted onto all hosts within the cluster. This file system contains files used to make udb changes. A cron job runs every five minutes on each host to look for these files and make appropriate udb updates.

To track user allocations, we use a database into which the accounting information from the pacct files is merged nightly. Data available is up to 24 hours out of date. Users have up to 24 hours in which they can exceed their usage allocations. We have chosen to allow this rather than incurring the overhead of more frequent updates.

As we have a heterogeneous cluster we need to accommodate different architectures. Our accounting deals only with CPU usage. To obtain a common currency between hosts, we multiply J90 CPU hours by 0.231 to obtain what we consider to be equivalent C90 hours. This calculation allows us to track usage after the fact.

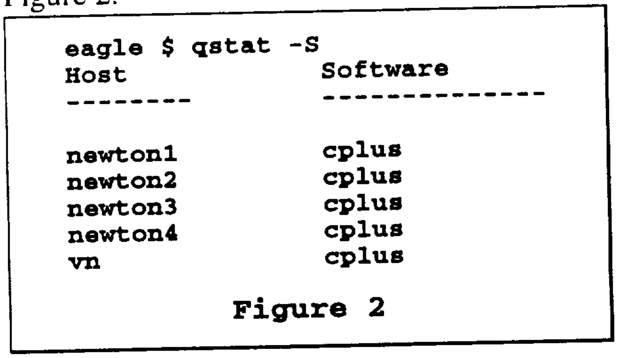
PBS Enhancements to Support Heterogeneous Clusters

To accommodate heterogeneity of the clusters clients, users are forced to specify CPU and wall time requirements in terms of C90 equivalents. The PBS client was modified to accept two new configuration variables. One variable is used to increase the amount of CPU time allowed for a job running on a J90. This same variable is used to convert the CPU time used to C90 equivalents. The second variable provides analogous functionality for wall time. Wall time indicates how long a job is allowed to run in order to complete its work.

An attribute was added which allows users to specify the host on which a job should run. Between the host attribute and the architecture attribute, users have the flexibility to choose which systems will be allowed to run their jobs. Consistent use of these attributes can limit the scheduler's ability to balance the system load across all hosts. However, the flexibility provided by the host attribute is important for data transfer jobs, for jobs used to debug code, and for jobs that deal with system administration tasks.

Previously, users would specify 16 CPUs and be prevented from running on some systems, or they would specify 12 CPUs and be unable to use all CPUs available to them. An attribute was added which allows users to specify the minimum number of CPUs they wish to use. This allows users to specify a range and to utilize all CPUs on a system without limiting themselves to specific hosts.

The software attribute was added to allow users to specify software available on some but not all systems, as listed in a configuration file. This allows the site to purchase software for some hosts without incurring fees to place the same software on all hosts. PBS's **qstat** was modified to list known differences as shown in Figure 2.



```
eagle $ qstat -S
Host              Software
--------          --------------

newton1           cplus
newton2           cplus
newton3           cplus
newton4           cplus
vn                cplus
```

Figure 2

```
eagle.sarita $ qstat -q
server: amespvp
Queue              Memory CPU Time Walltime Node Run Que Lm   State
---------------- ------ -------- -------- ---- --- --- --   -----
...
spcl               256mw 16:00:00    --      --    0   0  6    E R
batch              300mw 08:00:00    --      --   33  28 --    E R
hbatch             300mw 00:30:00 02:00:00   --    0   0  6    E R
rtime              256mw 03:00:00    --      --    0   0  3    E R
debug              300mw 00:05:00 00:30:00   --    0   0  9    E R
defer               48mw 02:00:00    --      --    0   0 --    D S
...
                                                  --- ---
                                                   35  59
=================================================================
interactive          10mw 00:10:00
```

Figure 3

```
eagle.sarita $ qstat -Aq
server: amespvp
Queue              Memory CPU Time Walltime Node Run Que Lm   State
---------------- ------ -------- -------- ---- --- --- --   -----
...
batch              300mw 08:00:00    --      --   31  28 --    E R
batch_vn           300mw 08:00:00    --      --    0   0 --    E R
batch_newton4       32mw 08:00:00    --      --    0   0 --    E S
batch_newton2      125mw 08:00:00    --      --    0   0 --    E S
batch_newton3       55mw 08:00:00    --      --    0   0 --    E S
batch_newton1      125mw 08:00:00    --      --    0   0 --    E S
batch_eagle         64mw 08:00:00    --      --    0   0 --    E R
                                                  --- ---
                                                   33  59
=================================================================
interactive          10mw 00:10:00
```

Figure 4

All the attributes mentioned were added to provide flexibility to users and sufficient information to schedule jobs appropriately. Much of the flexibility in scheduling is controlled by a multi-level queuing structure. Limit queues were added to the PBS software to provide a dynamic way of configuring queue limits on a host-by-host basis, while execution queues continue to limit the cluster as a whole. PBS's **qstat** was modified to provide this information. Figure 3 shows excerpts from the display showing cluster level limits. Figure 4 shows excerpts from the display showing some of the host level limits. Other limits also exist, such as number of CPUs and amount of space available to users on a solid state device (SSD, zero if there is none attached to the system).

Running the server on an Origin 200 while clients were on the C90s and the J90s also caused problems. While UNICOS has account ids, SGI uses project ids. Use of project ids was added to the server so that accounting information would reflect default accounts. Additionally, differences in word size resulted in incorrect conversions of attributes such as memory size. The software was modified to force the size of a word to be eight bytes. As mentioned in the "Reliability Improvements" section of this paper, modifications were also made to utilize SGI Failsafe.

A script keeps the scheduler and software configuration consistent across the hosts. This script allows the administrator to modify the configuration file on the active SGI Origin 200, sets file permissions appropriately, and submits jobs to PBS to push the updated file to all the clients.

User changes

To use the cluster effectively, users now need to be conscious of which home file system they want to access. Users also need to be aware of the host type when choosing which executable to run. Figure 5 shows a simple script as it existed on the J90 cluster prior to the existence of the heterogeneous cluster. In the script, $BIGDIR is a pointer to temporary space.

```
#!/bin/ksh
#PBS -lncpus=12
#PBS -lcput=4:00:00

cp -r data $BIGDIR
cd $BIGDIR
/u/username/a.out < input
cp -r * /u/username/newdata
```

Figure 5

To obtain the same behavior of the script in Figure 5 within the heterogeneous cluster, users simply have to modify their scripts to those shown in Figure 6 or in Figure 7. The script in Figure 6 would allow users to run on any Cray J90 with 12 or more CPUs. The script in Figure 7 would force the job to run on the specified host.

```
#!/bin/ksh
#PBS -lncpus=12
#PBS -lcput=1:00:00
#PBS -larch=j90

# J90 script

cp -r data $BIGDIR
cd $BIGDIR
/u/username/a.out < input
cp -r * /u/username/newdata
```

Figure 6

```
#!/bin/ksh
#PBS -lncpus=12
#PBS -lcput=1:00:00
#PBS -lhost=newton1

# J90 script

cp -r data $BIGDIR
cd $BIGDIR
/u/username/a.out < input
cp -r * /u/username/newdata
```

Figure 7

Figure 8 shows this same script updated to use the cluster effectively. The main differences are the use of **rcp** in place of **cp**, the range of CPUs allowed, and the architecture aware choice of executable.

```
#!/bin/ksh
#PBS -lmincpus=12
#PBS -lncpus=16
#PBS -lcput=1:00:00

rcp -r newton1:data $BIGDIR
cd $BIGDIR
if [ `uname -m` = "CRAY C90" ]
then
    /u/username/c90/a.out < input
elif [ `uname -m` = "CRAY J90" ]
then
    /u/username/j90/a.out < input
else
    echo "Unknown architecture"
    exit 1
fi
rcp -r * newton1:newdata
```

Figure 8

Most of our users choose a specific host. Due to memory requirements, there are jobs in input queues on our small C90 that would have started earlier on the large C90.

Researchers focus on the output results, not on the scripts used to generate them. When making changes, most users take the easiest path and simply add the host attribute to their scripts.

Our current system capacity exceeds our user requirements. Because of this, we see little impact from users specifying a host, except that the J90s remain primarily idle.

Possible Future Improvements

Work on the NAS cluster has resulted in several lessons for the future. To minimize user confusion and to minimize user script changes, a single home file system should be mounted via Failsafe or **cxfs**, if possible. To help ensure a balanced load across systems, users should be allowed to specify the host attribute only for debug and data transfer jobs. Administrators could still specify the host for system jobs. Users should continue to be allowed to specify the architecture attribute as this can impact code behavior significantly.