A Comparison of Job Management Systems in Supporting HPC ClusterTools

Presentation for SUPerG

Vancouver, Fall 2000

Chansup Byun and Christopher Duncan

HES Engineering-HPC, Sun Microsystems, Inc.

Stephanie Burks

University Information Technology Services, Indiana University

Abstract

This paper compares three most common job management systems and their workings with Sun HPC ClusterTools 3.1. Various aspects such as installation, customization, scheduling and resource control issues are discussed. The three chosen systems are: Load Sharing Facility (LSF), Portable Batch System (PBS) and COmputing in DIstributed Networked Environment (CODINE)/ Global Resource Director (GRD). We give a brief overview of each product but mainly focus on integrating these job management systems with Sun HPC ClusterTools. We provide useful guidelines to Sun HPC ClusterTools users when using job management systems with Sun HPC ClusterTools. We further demonstrate how to use the job management systems in support of commercial MPI applications with HPC ClusterTools.

1 Introduction

The Sun HPC ClusterTools software [1] is designed for the compute-intensive, technical computing environment. It provides an environment for the execution of serial and parallel high-performance applications on Sun servers as well as clusters of Sun servers. The Sun HPC ClusterTools also provides the software environment for developing and debugging applications that are parallelized and optimized for them. The Cluster Runtime Environment (CRE) in the Sun HPC ClusterTools provides a limited ability to distribute work load evenly among the servers.

However, in order to use computing resources efficiently and control them effectively, Sun HPC ClusterTools needs third-party "Job Management System" (JMS) software. There has been much JMS software developed for queueing, clustering and distributed computing systems. Kaplan and Nelson [2] list many of the earlier development efforts and compare them using various aspects of evaluation criteria. They provide an ancestry diagram of various JMS software. The diagram shows that "Network Queueing System" (NQS) is the first of its kind. NQS was first developed at NASA as a tool for scheduling batch jobs. NQS became the basis of the POSIX

1003.2d standard. Based on the POSIX standard, commercial and other public domain versions such as NQE (Network Queueing Environment), CONNECT: Queue, and PBS (Portable Batch System) were developed. Those versions enhanced NQS functionality such as load-balancing and higher-level resource control. Condor is another JMS software developed in the earlier JMS era. Condor influenced other JMS software such as LoadLeveler and CODINE. Condor is still actively developed and supported by the original developer, the University of Wisconsin.[1] More recently Papakhian [3] published a paper comparing some of the most popular JMS software from a user's perspective. She summarizes the three NASA studies to identify JMS software suited for NASA's job management requirement. The top three JMS software identified in the NASA study are PBS (Portable Batch System), LSF (Load Sharing Facility), and CODINE (Computing in a DIstributed Network Environment). CODINE was developed by Genias Software, which became Gridware[2] in 1999 by merging with Chord Systems. GRD (Global Resource Director), which has all the features from CODINE, was the result of a joint project of Genias Software, Raytheon and Instrumental. These three JMS software, LSF, GRD, and PBS, are selected in this study and investigated for compatibility in supporting Sun HPC ClusterTools 3.1 software.

We do not intend to compare these three JMSs in detail. Rather the paper is intended to overview the selected JMS products, outline how they work with HPC ClusterTools, identify problems and attempt to provide remedies for those problems. Some of the considerations are:

- Does the installation affect HPC ClusterTools installation?
- What kind of issues arise working with HPC ClusterTools?
- Can the JMS control parallel MPI applications?
- Are there any guidelines / best practices in using a particular JMS?

By doing so, this work will provide an overview of how to use these JMS products with Sun HPC ClusterTools, their problems and remedies. Users can jumpstart to use these JMS software with HPC ClusterTools with ease by following the suggestions made in this paper.

2 Overview of LSF, GRD, and PBS

LSF. "LSF is a suite of workload management products."[3] which can be purchased through Platform Computing Corporation.[4] Major features included with LSF are:

- Both an interactive and batch system
- Job scheduler and tool for analyzing cluster workloads
- Support for multiple UNIX variants and Windows NT
- Heterogeneous clusters
- NQS interoperability
- MultiCluster feature which allows for multiple LSF clusters to work together without the need to fully integrate them into one big cluster.

There also exist a functional API and other interfaces that allow a user or cluster administrator to tap into and to even extend LSF functionality.[4] For instance

1. www.cs.wisc.edu/condor
2. Recently Sun acquired Gridware, Inc.
3. quoted from "LSF Batch User's Guide", Platform Computing Corporation 1994-1997.
4. www.platform.com

in addition to the usual load indices of memory and cpu usage and others tracked by LSF, a site could add a new index say, for a special device or service. The LSF installation and documentation includes examples of this functionality along with examples of other ways to extend and use LSF.

Almost all of the functionality of LSF is available through graphical user interfaces (GUIs). Both user level commands and administrator commands have GUIs. The GUI commands are usually the command line names with "x" prepended. Many of the interactive and basic information commands begin with an "ls" (e.g. lsrun, lsinfo, lsid, lshosts, lsadmin, etc.) while the batch system commands begin with a "b" (e.g. bsub, bqueues, bhosts, badmin, etc.). There are also tools that are designed to work across the LSF cluster such as lsrcp, lstcsh, lslogin, etc.

GRD. In this study, we only used GRD since GRD is a super set of CODINE.The GRD has added two more extensions to CODINE: dynamic scheduling and resource policy management. However, CODINE should work similarly with Sun HPC ClusterTools since most of the features we tested are also found with CODINE.

The GRD is an advanced resource management tool for heterogeneous, distributed computing environments.[5] GRD provides advanced resource management and policy administration for UNIX environments composed of multiple shared resources. GRD has the following major capabilities:

- Dynamic scheduling and resource management
- Dynamic performance-data collection
- High level policy administration.

Dynamic scheduling insures that the most important work at any instant receives its deserved system share. Fine-grained policy definition supports the definition of a site-specific workload management strategy through the weighted combination of as many as four scheduling policies: shared-based, functional, initiation-deadline, and override schemes. Details of each scheme is explained in the GRD document. [5]

GRD provides both interactive and batch facilities. The interactive jobs can be submitted by three methods: qrlogin, qrsh and qmon. Qlogin is a telnet-like session. Qrsh is equivalent to the standard UNIX rsh command and replaces rsh and rlogin. With qmon, a user can initiate an interactive job, which will bring up an xterm window. The batch facility allows users to submit a job to GRD clusters. The batch job can be submitted through a command line input or from a batch script.

PBS. "The Portable Batch System, PBS, is a batch job and computer system resource management package. It was developed with the intent to be compliant with the POSIX 1003.2d Batch Environment Standard. As such, it will accept batch jobs, a shell script and control attributes, preserve and protect the job until it is run, run the job, and deliver output back to the submitter"[6] The PBS supports many UNIX variants and can support multiple systems grouped

5. quoted from "GRD User's Guide", Ver. 5.1, Gridware, Inc., 1993-2000.

6. "PBS Administrator Guide", Release 2.2, MRJ Technology Solutions, Feb. 18, 2000. Now MRJ is part of Veridian Systems.

together into a cluster. Its features include:

- Both an Interactive and Batch system
- Automatic Load-Leveling
- Job-interdependency between batch jobs
- Fully Configurable Scheduler
- Automatic File Staging
- Security and Access Control Lists

At present, PBS is a source file level distribution, available from Veridian Systems, that is built by the end user. This is a "feature" that stands out in comparison to the other JMS products discussed in this paper.

The PBS Scheduler is implemented in a modular fashion that allows a site to develop complex scheduling polices using the C language, Tcl scripting language or PBS's C language extensions BaSL, which stands for Basic Scheduling Language.[6]

Along with the standard commands to launch and monitor jobs (qsub, qstat, etc.) there are administration commands for modifying the queues and administering them (qmgr, qenable, qdisable, etc.). There is an option to build a Tcl/Tk interface called xpbs that gives you GUI access to the PBS system.

3 Installation and Cluster Construction

As with any installation it is beneficial and a likely time saving to read the installation guide before starting the installation. Even a cursory look will indicate possible issues and things to think about before installing. In this section, we summarize the default installation of each product for completeness and point out some issues and helpful tips during the installation.

LSF. The LSF 3.2.4 distribution[7] we used came in multiple compressed tar files. The 64-bit and 32-bit version are separate so you will have to decide which version to load depending on your OS setup. Once you have decided this you can uncompress and untar the distribution and you are almost ready to start installation. It is best to first create an administration account such as lsfadmin. By creating this account you can allow for LSF administration without having to give full root access to the administrator. If you have non-root administration accounts such as "staff" one of these may also be a good choice.

At the top level of the untarred distribution is the install and host setup command lsfsetup. This script and its utilities do most of the underlying work for you. It will ask you questions about the setup and, once answered, will install LSF onto the system. You may find it easiest to install the LSF distribution into an NFS mounted directory since this will make adding nodes easier. One could view the install as two phased: installing the binaries and files into an accessible area then setting up the individual hosts.

The answer to first question from lsfsetup should be to choose "Custom Install". If you are going to use Sun HPC with LSF you will want both the LSF Batch and Parallel options, the later being required by the Sun HPC plugins. You may load other LSF products also but LSF Parallel must be available.

Once LSF is installed in an accessible directory available to each node you will need to setup each host. This will add it

7. For the Sun HPC 3.0 release Sun distributed LSF and released it as the package SUNWlsf. Sun is no longer an LSF reseller.

to the cluster and create and modify some configuration files. Setting up each host can be done through the lsfsetup command. Once each host is setup, you may either configure some queues through the command line interface lsadmin and badmin or startup the GUI xlsadmin. If you have previous LSF experience you could even edit the text configuration files directly.

One key file that is placed on each node is /etc/lsf.conf. This has a large number of variable definitions that the LSF commands will use to find out about the installation.

GRD. The GRD installation requires a few things to be prepared before the installation. First, the installation needs an administrator account. The administrator can be an existing administrative login or a new login such as "grdadmin". This account will own all of the files in the GRD installation and spooling directories. The administrator can configure and administer the cluster once it is installed. This user should not be "root" in order to avoid a problem to configure root read/write access for all hosts on a shared file system. Second, the installation root directory ($GRD_ROOT) should be created and the distribution sources unpacked under the installation root directory. This should be done by the administrator so that all the files are owned by the administrator. Also the installation requires a valid license key.

The next thing is to configure TCP communication for GRD daemons. All hosts in the cluster must use the same port number. The port number can be placed in several places such as NIS services or NIS+ database or /etc/services on each machine as shown below.

```
grd_commd 535/tcp \
# communication port for GRD
```

A default GRD cluster consists of one master host and an arbitrary number of execution hosts. The master host controls the overall cluster activity while the execution hosts control the execution of the jobs being assigned to them by the master host. A single host may concurrently act as a master host and as an execution host.

The installation should be done on the master host first and then be done on the execution hosts in arbitrary sequence. This installation requires the “root” access to install all available GRD features. The installation scripts, install_qmaster and install_execd, are in the GRD root directory. Both installation scripts will ask questions about the cluster configuration and then will install the respective GRD part upon answering all the questions.

With the successful installation, the following GRD daemons should be running on the master and execution hosts.

```
grd_qmaster (master host)
grd_schedd (master host)
grd_execd (execution hosts)
grd_commd (all hosts)
```

The GRD daemons can be started and stopped by the following script created during the installation.

```
$GRD_ROOT/default/common/grd5
```

An appropriate environment setup should be done before using any of GRD commands. This setup can be done with the following setup file created with the installation.

```
% source $GRD_ROOT/default/ \
common/settings.csh
```

This file sets correct paths for all GRD commands and their man pages. The administrator can modify or reconfigure the default setup using "qmon" graphical user interface.

It should be noted that GRD doesn't allow the "stty" command in personal shell resource files such as ".cshrc", ".login", or ".profile". Users should not execute the stty command when a batch job is submitted otherwise the batch job will exit immediately.

PBS . The source distribution version of PBS we had available was 2.2p11 which unfortunately only supports 32-bit on Solaris. A future version (2.3) should have 64-bit support for Solaris. We had no problems compiling the source using the Sun Workshop 5.0 C compiler.

Once the source distribution is untarred you can run the configure command, build PBS and then install it. There are many options available for the configure phase and you should consult the manual and decide which is best for your site. Some of the options may be very familiar from other open source distributions that use a similar configure command.

The PBS daemons will need to be run as root on each system. There are three daemons: pbs_server and pbs_sched, which are started on only one system per cluster, and the pbs_mom, which should be started on all nodes where jobs will be executed. There is a file which lists all the systems in the cluster that can be made available for jobs, `$PBS_HOME/server/priv/nodes`.

After the daemons are started the qmgr command can be used to configure queues and you can then start submitting jobs. There is the option to have PBS administrators and these can be listed in the file: `$PBS_HOME/server/priv/acl_svr`.

4 Compatibility with Sun HPC ClusterTools

During the development of Sun HPC 3.0 Sun worked with Platform Computing Corporation to create plugins (shared object libraries) that would allow Sun HPC Clustertools jobs to be launched and controlled directly by LSF. These plugin libraries add the extra parallel support and features that Sun HPC Clustertools requires. In this regard LSF stands in sharp contrast to the other JMS in this report by being tightly integrated with Sun HPC ClusterTools.

LSF. The features and functionality available for a standard LSF job are available to Sun HPC Clustertools jobs launched with LSF.

For Sun HPC to install properly with LSF, LSF should be installed first. There is also the requirement to get a special Sun HPC Clustertools patch from Platform Computing Corporation which will make minor modifications to your LSF installation. Without this patch LSF 3.2.4 and Sun HPC ClusterTools 3.1 will not work properly.

Once LSF is installed and the special LSF patch is applied you can install Sun HPC ClusterTools. The Sun HPC installation configuration file should have the options:

```
LSF_SUPPORT="yes"
MODIFY_LSF_PARAM="yes"
```

You will also need to specify the LSF_CLUSTER_NAME in the HPC install configuration file. The Sun HPC ClusterTools installation scripts are aware of LSF and will make the proper modifica-

tions during installation. The interactions between Sun HPC Clustertools and LSF are fairly seamless after installation. It should be noted that, with this Sun HPC ClusterTools installation, the CRE commands will not be available. Instead of using `mprun` to launch jobs you use the native LSF `bsub` command.

After Sun HPC Clustertools is installed two queues named “hpc” and “hpc-batch” are configured to run Sun HPC Clustertools jobs. If you wish to add other queues for MPI jobs take note of the variables set for these queues in the lsb.queues file. Of particular note is the “JOB_STARTER = /opt/SUNWlsf/bin/pam -t” entry which is especially critical for MPI jobs.

One should be careful to not use the LSF patched mpicc and mpif77 commands for compiling since those are intended for other MPI distributions and LSF operation. Signaling, job resource usage and other issues follow the normal LSF job interaction. The Sun HPC Prism debugger and visualization tool is available with full functionality.

GRD. The GRD system provides a "Parallel Environment" (PE) to interface with parallel jobs such as multi-threaded and message-passing applications. The administrator should set up appropriate start-up, stop and signaling procedures to make sure that those parallel jobs are successfully administered through the GRD system. The current GRD does not have a tightly integrated interface with the Sun HPC ClusterTools. However, the GRD allows the user to submit and execute message-passing applications under the CRE of the Sun HPC ClusterTools with minor modifications in the start-up script. The output shown in Fig. 1, which is from the "diff" command on the original and modified start-up

```
 % diff startmpi.sh startsunmpi.sh
6c6
< # usage: startmpi.sh [options] <pe_
hostfile>
---
> # usage: startsunmpi.sh [options] <pe_
hostfile> <mprun_path>
27c27,28
< echo $host
---
> # Add a unit after hostname for mprun
-Mf format
> echo $host 1
79,81c80,82
< # ensure job will be able to exec
mpirun
< if [ ! -x $MPIR_HOME/util/mpirun ];
then
<    echo "$me: can’t execute $MPIR_
HOME/util/mpirun" >&2
---
> # ensure job will be able to exec
mprun
> if [ ! -x $MPIR_HOME/mprun ]; then
>      echo "$me: can’t execute $MPIR_
HOME/mprun" >&2
```

Figure 1: Modification for Sun MPI

scripts, shows changes made to support CRE of HPC ClusterTools. It should be noted that the modification introduced in Fig. 1 is only applied to $fill_up option, which is explained later in this section.

When configuring a PE, the GRD has a number of options to specify the allocation of CPUs for parallel jobs: positive integer, $pe_slot, $fill_up, and $round_robin. Both integer and $pe_slot limit the number of processes to be spawned within an execution host. However, when a unit number is specified for the allocation rule, only one process is allowed on a host, but the process can be spawned on other hosts as well. This is a useful option for spawning a single process on each host or a multi-threaded MPI application. Both $fill_up and $round_robin options are useful for spawning MPI processes across execution hosts. The $fill_up enforces to fill up

the available slots of an execution host before using another host whereas the $round_robin distributes processes among the execution hosts as evenly as possible by filling up the one available slot of each execution host sequentially.

As far as resource control is concerned, when an MPI application is signaled due to the enforced resource limit, in some cases, some of the MPI processes may be left lingering around. In order to eliminate those zombie processes, it is necessary to define a terminate_method on a per queue basis or to modify the stop script for a PE. One easy example might be to use "pkill -9 -u <user>" command. However, this command will kill all other processes owned by the user on the same host. In order to avoid such a problem, more sophisticated procedure has to be developed. The CRE command, "mpps", can provide useful information about MPI jobs. This command output can be used to build a better stop script for the lingering MPI processes.

PBS. Although PBS has an API called Task Manager which will launch parallel tasks, there is no clear way to integrate its functionality with Sun HPC's CRE and further discussion goes beyond the scope of this paper.

For a given Sun HPC Clustertools job one can submit a PBS job that then calls mprun to execute the job in parallel. Unfortunately PBS will only be aware of the mprun process and the actual MPI processes of the job are not known to it. Thus monitoring the job and its resource usage is not available. Since resources are not enforceable it is up to the users to stay within the resource limits specified by the job. The following is an example of an 16 process job, requested for 2 nodes (8 cpus each) launched from PBS and passed to mprun:

```
% qsub -l nodes=2:ppn=8
/opt/SUNWhpc/bin/mprun -np ‘wc -l
$PBS_NODEFILE | nawk ’{print $1}’’ -
Mf $PBS_NODEFILE a.out
```

This will request that mprun use the same host placement for the processes as PBS has chosen.

One could also specify the required resources, such as memory or disk usage, to the PBS qsub command which could then block the job from starting if there would be oversubscription. However since PBS is still unaware of the actual processes involved in the parallel job the resource limits will not be enforced except for wall-clock time.

Unfortunately almost all of the PBS resource usage tracking and enforcement functionality is lost due to the disconnect between the PBS job and the processes launched by the CRE mprun command. PBS expects a job's processes to remain in the same UNIX session id, which the mprun process certainly cannot satisfy, and there is no reverse registration capability for mprun to register the processes of the job to PBS.

Resource usage enforcement is an important feature for most sites. One option that can be explored to improve the situation is writing scripts and utilities that will collect the resource usage for Sun MPI jobs and then check these against the resources requested by the job. Using the `mpps` command with the `-p` option will help in tracking down the individual processes of a job. One could then combine output from the `ps` and `qstat` commands and cross reference the jobs usage and allocation. Though this is not the best solution it may help to bridge the gap of resource allocation and enforcement.

5 Interaction with MPI Applications

LSF. Since Sun HPC Clustertools has the ability to be tightly integrated with LSF through the Sun RTE plugins there are no special interactions with MPI applications. One issue that may arise is that some independent software vendor (ISV) applications may be written to assume that the Sun CRE is available and they may reference mprun commands. You may wish to contact the ISV directly to inquire about the correct changes that could be made to allow the application to work with LSF. It may also be possible to find the parts of the application that call the Sun CRE commands (such as mprun) and modify those yourself. Other than possible ISV application issues all the features and functionality of LSF parallel jobs exist with Sun HPC MPI applications.

GRD. The GRD provides a "parallel environment" to execute an MPI job. In order to run an MPI job, the “-pe” option should be used as shown below when submitting the job.

```
#$ -pe <PE_name> <CPUs_requested>
```

The batch script to run an MPI job is also shown below. Each of the GRD directives begins with #$. In this script, `$GRD_O_WORKDIR`, `$NSLOTS`, and `$TMPDIR` are inherited from the GRD system. "HPC" is the name of the selected PE environment and 4 CPUs are requested. Thus `$NSLOTS` will have the value 4. The `$TMPDIR`/machines file contains the allocated host and process information, which created by the start-up script described in the previous section.

```
#!/bin/sh
#$ -cwd
#$ -M user@foo
#$ -m es
#$ -o monte.out
#$ -e monte.err
#$ -N monte
#$ -pe HPC 4
cd $GRD_O_WORKDIR
/opt/SUNWhpc/bin/mprun -np $NSLOTS \
-Mf $TMPDIR/machines ./monte1
```

Figure 2: GRD batch script

The GRD seems to provide reasonably accurate accounting information for wallclock time, CPU time, memory and number of CPUs when all processes are assigned to within an execution host. However, it still fails to report other usage information such as user time, system time and IO information. Furthermore, it does not report CPU time and memory information correctly when the job is distributed among two or more hosts.

It is also noted that CPU limit control does not work as documented when working with MPI applications. For example, the CPU time limit on the job works as if the limit is imposed on each MPI process. Some other resource limits such as wallclock time were able to signal the job when it reached the user specified limit but some MPI processes were still lingering around. In this case, a proper stop script was required to clean up the remaining MPI processes as described in the previous section.

During an application test using MSC NASTRAN for the GRD with Sun HPC ClusterTools, we noticed that there was a conflict in resource allocation. NASTRAN has its own way to distribute MPI processes, based on a round robin scheme, among the participating execution hosts. The process order may not correspond with the one specified by the GRD. If the allocation is done within a single execution host, there will not be such a problem.

In order to accommodate various resource allocation schemes allowed in the PE, one may define many PEs, one PE for each allocation type, for the same cluster of machines. Then, a user can choose a proper PE depending on the resource allocation requirements.

PBS. Submission to PBS is best accomplished using a script like the two examples shown in Fig. 3. Each of the PBS Directives begins with #PBS and is followed by one or more qsub options.

```
node% cat sun_A_4.pbs
#!/bin/csh
# Timeshare PBS submission script
#PBS -l ncpus=4
#PBS -l mem=2gb
#PBS -A user
#PBS -M user@foo

/opt/SUNWhpc/bin/mprun -np 4 \
/stburks/sun.A.4 >
/stburks/sun.A.4.txt

node% cat sun_A_4c.pbs
#!/bin/csh
# Cluster PBS submission script
#PBS -l nodes=4:ppn=2
#PBS -l mem=2gb
#PBS -A user
#PBS -M user@foo

ncpus=‘wc -l $PBS_NODEFILE | \
nawk ’{print $1}’‘

/opt/SUNWhpc/bin/mprun -np $ncpus \
-Mf $PBS_NODEFILE /stburks/sun.A.4 \
> /stburks/sun.A.4.txt
```

Figure 3: PBS batch script for Timeshare and Cluster

Any job requirements that are not specified in the script will default to the limits defined for the server.

A PBS interactive job is submitted either by including the -I with the qsub command on the command line or by including the -I as a #PBS directive in a script file.

MPI jobs will be scheduled appropriately by PBS. However, with the exception of wallclock time, the resource usage will not be tracked. Once the wallclock time for the queue has been exceeded, the job will be terminated. Although PBS is unaware of the nodal processes owned by the CRE, the processes will terminate, should the mprun process tracked by PBS be terminated.

6 Guidelines and Best Practices

In this section, we summarize what we have discussed in the previous sections for using JMS products in working with Sun HPC ClusterTools 3.1. We like to note that Hassain [7] published best practices for Sun HPC ClusterTools, which provides useful information for using Sun HPC ClusterTools.

LSF. When configuring LSF for a site the follow suggestions are made:

- Install the LSF distribution onto a network file system such as NFS that is accessible to all the nodes you wish to have in the cluster.
- Either create an LSF administrator account directly or specify one of your non-root administrator accounts as the LSF manager during install.
- Be sure to have LSF 3.2.4 installed, with the Sun HPC Clustertools patch (available from Platform Computing) before you install Sun HPC Clustertools. Otherwise the Sun HPC Clustertools installation will not properly configure itself with LSF.
- As with any piece of complex software, carefully reading the documen-

tation is likely to save a lot of time and effort.

GRD. To make the best use of the GRD system in working with Sun HPC ClusterTools, the following suggestions should be made in advance.

- Configure the PE appropriately according to the users's needs. The PE can support both multi-threaded and message-passing applications. Depending on the parallel application, a PE has to be configured differently. For example, the allocation rule for an multi-threaded application should be "$pe_slots", which enforces allocation of processes within an execution host. However, the allocation rule for MPI applications can be other rules such as "$round_robin" or "$fill_up".

- Create PEs suitable for each allocation scheme for the same cluster and choose a proper one when submitting an MPI job.

- Modify start-up and stop scripts as explained and make sure that the scripts work with MPI applications of interest.

- Be aware of any MPI application package allocating computational resources by itself. For example, MSC NASTRAN allocates CPUs in a round-robin fashion among the given hosts by an internal launching script before actually issuing the "mprun" command in its script. This allocation probably doesn't match with the GRD resource allocation because it doesn't use `$TMPDIR`/machines shown in Fig. 2. The GRD will report false information.

PBS. For a site which is using PBS the following may be helpful:

- When configuring the source distribution, be careful to review the configure options before building. You may wish to change the PBS_HOME directory by setting --set-server-home=DIR.

- There may be better solutions or workarounds found to bridge the gaps between PBS and Sun HPC Clustertools. Stay tuned.[8]

7 Summary & Conclusion

This paper overviews three job management systems, GRD, LSF and PBS and their compatibility with the Sun HPC ClusterTools 3.1 software. We provide useful tips and guidelines in using these JMS systems with HPC ClusterTools 3.1. and limitations of each JMS product in using the Sun HPC ClusterTools. At present, LSF is tightly integrated with the Sun HPC ClusterTools whereas both GRD and PBS are loosely integrated with the Sun HPC ClusterTools. Thus, LSF provides complete control of MPI applications while both GRD and PBS has limited support and control of MPI applications. A tight integration with CODINE/GRD will be available in the future.

Acknowledgements

The first author would like to thank Omar Hassaine for suggesting the idea for this paper.

8. pbs-users@pbspro.com

References

[1]Sun HPC ClusterTools 3.1 Documentation set, http://docs.sun.com.

[2]Kaplan, Joseph A. and Nelson, Michael L., "A Comparison of Queueing, Cluster and Distributed Computing Systems," NASA TM 109025, NASA Langley Research Center, June 1994.

[3]Papakhian, Mary, "Comparing Job-Management Systems: The User's Perspective," IEEE Computational Science & Engineering, April-June 1998.

[4]LSF 3.2 Documentation set, http://www.platform.com

[5]GRD Installation and Administration Guide, Ver. 5.1, Gridware, Inc., San Jose, California

[6]PBS Documentation Set, http://www.pbspro.com

[7]Hassaine, Omar, "HPC ClusterTools Best Practices," SUPerG meetings, Paris, April 2000.

Copyrights