SUN BLUEPRINTS APPROVED

HPC Administration Tips and Techniques

Omar Hassaine, CPR Engineering-HPC

Sun BluePrints™ OnLine—Oct 2002

Sun microsystems

http://www.sun.com/blueprints

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95045 U.S.A.
650 960-1300

Part No.817-0079-10
Revision 02, 08/29/2002
Edition: October 2002

DOCUMENTATION IS PROVIDED “AS IS” AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

CETTE PUBLICATION EST FOURNIE “EN L'ETAT” ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REPONDRE A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.

HPC Administration Tips and Techniques

This article contains a brief introduction to the features introduced with the latest Sun HPC ClusterTools™ 4 software and discussions of the administration practices for successfully configuring the Sun HPC ClusterTools software. The first administration practice covered in this paper has long been requested by HPC customers and deals with the ability to provide root-privileges to regular HPC users to maintain the Sun HPC ClusterTools software. The second practice relates to configuring mixed HPC clusters. This article also introduces the latest release of the Sun™ Grid Engine (Sun GE) software release 5.2.3.1 and the Condor standalone user-level checkpointing library. Best practices are given on how to configure a checkpointing and migration environment by using both Sun GE software and the Condor standalone checkpointing libraries.

Introduction to Grid Computing

The products covered in this paper are among the basic and fundamental components needed to build a grid infrastructure. Grid computing has been making the headlines lately and is touted as the new computing paradigm for this decade because it can increase the return on your computing assets by more effectively using your existing hardware. The Sun GE software handles compute and resource management at the cluster level by providing the required hooks to access the computing grid through known application program interfaces (APIs), such as Globus and Avaki. The Sun HPC ClusterTools 4 software provides the distributed parallel programming environment that enables users to execute their message passing interface (MPI) programs on a Sun UltraSPARC™ based cluster. The Condor standalone libraries allow serial threaded programs to be checkpointed for later restart if the need arises.

Sun HPC ClusterTools 4 Software Overview

The Sun HPC ClusterTools 4 software is designed specifically for compute-intensive, technical computing environments and enables the execution of serial and parallel high-performance applications. It provides middleware to facilitate and manage a workload of highly resource-intensive applications on Sun servers, as well as clusters of these servers. Additionally, it provides the software development environment for creating and debugging MPI applications that are parallelized and optimized for Sun servers and clusters.

The Sun HPC ClusterTools 4 software is the follow-on to the Sun HPC ClusterTools 3.1 release. Both versions can be installed on the same system, but only one release can be activated for use by using a `reconfig` command. The Sun HPC ClusterTools includes the following new features:

- Cluster nodes can span over subnets.
- Administrators can use the `sudo` utility to set superuser (root) privileges.
- The software has been optimized for better visualization.
- Loadable protocol modules are supported.
- UltraSPARC III processors are supported.
- The next-generation high-performance interconnect is supported.

The Sun HPC ClusterTools 4 software supports the Solaris™ 8 Operating Environment. It is also released under the Sun Community Source License. The Sun HPC ClusterTools 4 software consists of the following components:

- Cluster runtime environment
- Sun message passing interface
- MPI I/O
- Sun Prism™ software programming environment
- Scalable Scientific Subroutine library (S3L)
- Parallel file system (PFS)

FIGURE 1 shows the software architecture of the Sun HPC ClusterTools 4 software.

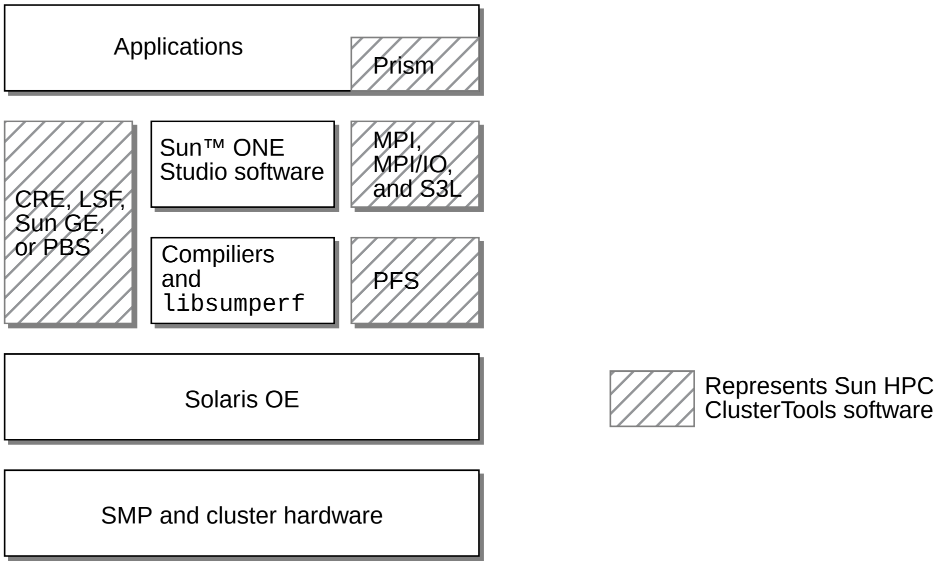


FIGURE 1 Sun HPC ClusterTools Software Architecture

The Sun Cluster Runtime Environment (Sun CRE) is a principal component of the Sun HPC ClusterTools 4 software because it provides the job launching and load balancing capabilities for MPI-based C, C++, and Fortran programs. The Sun HPC ClusterTools 4 software supports up to 2048 processes and up to 64 nodes in a cluster. The software also supports Platform Computing's load sharing facility (LSF) as a distributed resource manager. The Sun GE software supports only the external launcher mechanism of MPI programs and is loosely integrated with the Sun HPC ClusterTools 4 software (refer to the Sun HPC ClusterTools 4 software documentation). The LSF provides batch queuing capabilities and integrated launching of MPI applications. A fully Sun CRE integrated, portable batch system (PBS) was recently made available through a patch. FIGURE 2 shows the current distributed resource management integration types with the Sun CRE software.

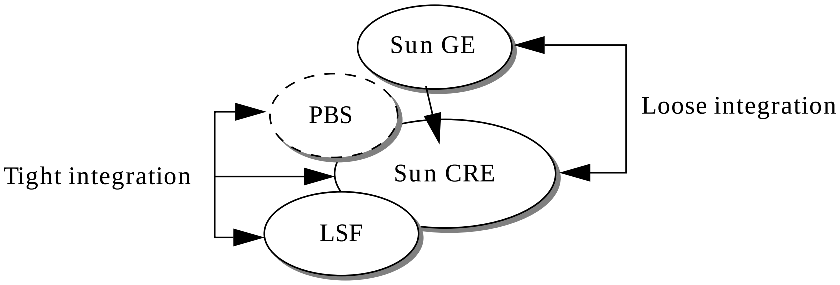


FIGURE 2 Distributed Resource Management Software Integration With CRE

Message Passing Interface

The Sun message passing interface (Sun MPI) component is an optimized version of the industry standard message passing interface communication library. The following list contains the main features of the Sun MPI library:

- Multithreaded programming
- 64-bit safe, thread safe, and trace normal form (TNF) versions of the library
- Multiple protocols
- Clusters of symmetric multiprocessing machines (64 nodes and 2048 processes)
- Sun remote shared memory (RSM) protocol over the next-generation Sun interconnect
- MPI-2 standard compliant
- One-sided communication within a single node
- `MPI_Comm_spawn` functionality of the MPI-2 standard to support the multiple-program multiple-data (MPMD) model

MPI I/O

The Sun MPI I/O software provides parallel I/O capabilities for message passing programmers and supports all of the MPI I/O routines defined in the MPI-2 standard. The Sun MPI I/O API also supports both UNIX™ file systems and Sun™ PFS software.

Sun Prism™ Software Graphical Programming Environment

The Sun Prism software environment currently provides support for the following major features:

- Multiprocess and multithread programs in 32-bit and 64-bit environments
- MPI performance analysis using the Solaris Operating Environment trace normal form
- Data visualization
- Debugging of MPI codes that use the `MPI_comm_spawn` functionality of the MPI-2 standard

Scalable Scientific Subroutine Libraries

The Sun scalable scientific subroutine libraries (Sun S3L) is a thread-safe parallel math library that contains a set of parallel and scalable routines widely used in scientific computing. The Sun S3L uses the Sun performance library (`libsunperf`) and provide local and global utilities for MPI applications. The performance library is part of the Sun WorkShop™ software development tools. The functions added in the Sun HPC ClusterTools 4 software (Sun S3Ls version 4.0) are as follows:

- Additional solvers and utilities for sparse systems
- Support for linear programming and optimization
- Functions for calculating equity option pricing
- Additional transforms (Walsh, sine, cosine)
- Support for additional subroutines (Cholesky, QR)
- Optimizations for UltraSPARC III processors

Sun PFS Software

The Sun PFS software is a parallel file system designed to support the parallel I/O found in HPC programs. It is also transparently supported by MPI I/O. Refer to the Sun HPC ClusterTools 4 documentation for more information.

Using the sudo Utility to Configure Non-Superuser Privileges

The administration of the Sun HPC ClusterTools 4 software should be minimal after a successful installation. There are, however, a few best practices that should be followed to help ease the HPC administration task. Configuring non-superuser (`root`) privileges is one of these practices.

System administrators can allow non-superuser users who are HPC knowledgeable and who are using the Sun HPC ClusterTools 4 software to perform administrative tasks without the need of superuser privileges and without the possibility of inadvertently affecting other systems in their network. The publicly available `sudo` software (see "References" on page 20) meets this requirement by allowing specified superuser-only commands to be executed on specific hosts by properly configured users.

The `sudo` software package has three different components that are of particular importance:

- `sudo`(8) command
- `visudo`(8) command
- `sudoers`(5) file

Note – The order of installation of these packages does not matter.

The manual pages for the above components provide further details. For the `sudo` command to allow non-superuser administrators to install and administer the Sun HPC ClusterTools 4 software, the `/etc/sudoers` file needs to be edited and configured using the `visudo` editor command (see "References" on page 20 for a link to the Sun GE software project site that contains a sample `/etc/sudoers` file). It is important to note that `vi`(1) or other editors cannot modify the contents of this crucial file. It is also important to know the path to the various HPC related commands and their expanded links. For example, the `mpadmin`(1M) command is a symbolic link that points to a shell script (`../isa.sh`), taking the name of the symbolic link as an argument and running a similarly named executable in an architecturally specific directory, such as the `sparcv9` directory. The following list contains the specific HPC related commands affected by `sudoers` file:

- `mpadmin`
- `mpinfo`
- `mpkill`
- `mpps`

- pfsstart
- tm.mpmd
- tm.omd
- tm.rdb
- tm.spmd
- tm.watchd

See “Appendix” on page 20 to see the main part of the /etc/sudoers file that needs to be modified to allow regular users to administer the Sun HPC ClusterTools 4 software on specified HPC hosts. After the sudoers file modifications are made, there is no need to reboot the system or to take any other action for the new changes to take effect. To test the validity of the changes, an eligible user needs only to run the sudo(8) command followed by the desired HPC command on the host(s). It is recommended to use the latest version of the sudo software (version 1.5.9 or a subsequent release) because an earlier version would not accept the wildcard character (*) and prevent the setup as defined by the /etc/sudoers file.

Running Multiple Releases of the Sun HPC ClusterTools Software

A need may arise where a production HPC cluster is using an earlier version of the Sun HPC ClusterTools software and a newer release of the Sun HPC ClusterTools 4 software is being transitioned. This configuration requires the use of the NFS install option and configuration of the NFS server to be outside of both clusters. The Sun HPC ClusterTools 4 software that serves each cluster needs to be installed in a separate file system so that the HPC related configuration files cannot be confused by the two software releases.

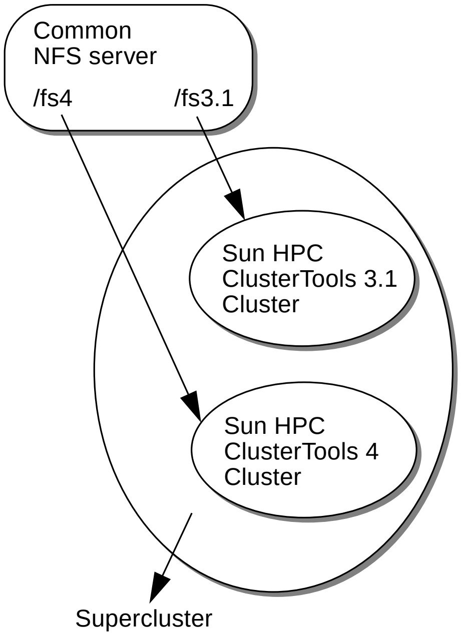


FIGURE 3 Mixed Cluster Configuration

FIGURE 3 shows a two-cluster configuration. One cluster is running the Sun HPC Cluster Tools 3.1 software, and the other is running the Sun HPC ClusterTools 4 software. The `/fs3.1` file system serves the 3.1 cluster, and the `/fs4` file system serves the Sun HPC ClusterTools 4 software cluster. Note that there can be more than two clusters served by the same NFS server, as long as the rule of one separate file system per cluster is observed.

The Sun HPC ClusterTools 4 software now supports the case where a partial cluster is running Sun HPC ClusterTools 4 software that is part of a large cluster configured with another distributed resource management software. For example, a customer site could be using Platforms Computing's load sharing facility (LSF) to manage a large cluster of nodes, and a portion of this cluster is running the Sun HPC ClusterTools 4 software, together with LSF.

User-Level Checkpointing Migration

Checkpointing is one of the hottest features that HPC sites request from vendors because a lot of HPC codes run for hours and even days, and an interruption during their execution should be able to save the state of the run for later resumption without starting over from the beginning. There are three types of checkpointing:

1. Kernel level

2. User level, using checkpointing libraries

3. Application level

The kernel level checkpointing is provided by the native operating system. The Solaris OE does not presently provide this capability. Application level checkpointing is provided from within the application by adding specific code that allows the application to checkpoint itself. In this paper, we focus on the remaining type of checkpointing, which involves publicly available user-level checkpointing libraries.

Sun™ Grid Engine Software

The Sun Grid Engine (Sun GE) software, previously know as CODINE, is a distributed resource management software that allows sites to efficiently manage and use the compute resources of machines across their network.

The Sun GE software is available free for download from the Sun Grid Engine website:

`http://www.sun.com/gridware`

The Sun GE source code has also recently been released for the open source public community at the following site:

`http://gridengine.sunsource.net`

The Sun GE product is Sun's distributed resource management tool for cluster grids. It is responsible for managing and submitting jobs to available compute resources in an individual grid. The Sun GE software maximizes CPU utilization, increasing productivity and return on investment. An enterprise edition version of the Sun GE software, Sun Grid Engine, Enterprise Edition (Sun GEEE), is Sun's resource management software solution particularly targeted at enterprise grids. This new version orchestrates and delivers computational power according to enterprise policies that are set by the organizational technical and management staff. The Sun GEEE software uses these policies to examine the available computational resources within the enterprise grid, then gathers, allocates, and delivers these resources automatically so that highly optimized resource usage is achieved across the enterprise grid. A controlled share of the total computing resources is assigned to groups, users, and departments by using the Sun GEEE software.

Condor Checkpointing Library

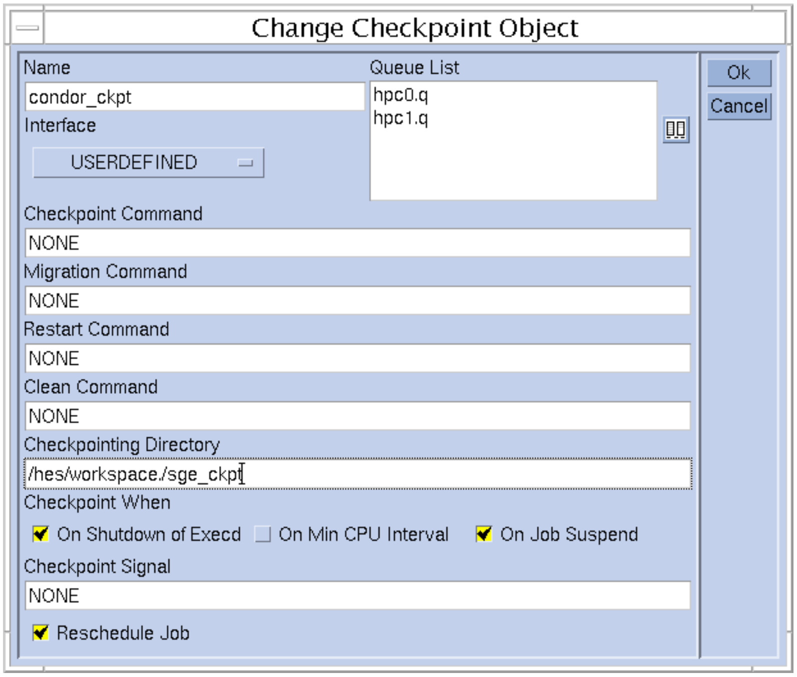
The Condor project at the University of Wisconsin is a fully-distributed resource management software project, including a user-level checkpointing library. This library can be used either as an integral part of the Condor system or as a standalone part of another distributed resource management software, such as the Sun GE software product. This article includes descriptions of the standalone library that is used with the checkpointing facility in the Sun GE software.

Sun GE Software Checkpointing Environment

In a Sun lab experiment, the following system configuration was used to test the Sun GE software checkpointing environment:

- Two Sun Ultra™ 80 workstations were used. One was configured as master-submit-execution host and the other as an execution host. Both machines were loaded with the Solaris 8 OE, the Sun GE 5.2.3.1 software, the Forte™ 6U1 compilers, and the Condor 6.2.1 libraries.
- The two Sun GE host queues were configured to support checkpointing.
- The application was set up to checkpoint when the `sge_execd` daemon was shut down or when the job was suspended.
- Sun GE was configured so that the job would be rescheduled in case it was suspended.
- The checkpoint signal was set up to `SIGTSTP` because the Condor libraries use it to checkpoint the application. Alternately, there was the `SIGUSR2` signal used by the Condor libraries to checkpoint the application and then continue its normal execution.

- Finally, *user-defined* checkpointing was set. User-defined checkpointing means that the application periodically writes checkpoints without any intervention by the Sun GE software. At restart time, the application continues from the last checkpoint. FIGURE 4 shows the Checkpoint Configuration window used during the test.



Name
condor_ckpt
Interface
USERDEFINED
Queue List
hpc0.q
hpc1.q
Checkpoint Command
NONE
Migration Command
NONE
Restart Command
NONE
Clean Command
NONE
Checkpointing Directory
/hes/workspace./sge_ckpt
Checkpoint When
☑ On Shutdown of Execd □ On Min CPU Interval ☑ On Job Suspend
Checkpoint Signal
NONE
☑ Reschedule Job

FIGURE 4 Checkpoint Configuration Window in the `qmon` GUI

Standalone Checkpointing Setup

The full Condor source code can be downloaded as a TAR file from:

`http://www.cs.wisc.edu/condor`

For the Sun lab experiment, there was no need to install the whole Condor software because only the entire `lib` subdirectory and the `condor_compile` command from the `bin` subdirectory was needed.

The condor_compile shell script needs to be modified at the following line:

```
CONDOR_LIBDIR=‘condor_config_val LIB‘

to

CONDOR_LIBDIR=full_path_of_lib
```

Where *full_path_of_lib* is the path to the highest level of the Condor lib subdirectory.

The above setup allows sequential applications to be checkpointed by using the user-level checkpointing Condor libraries.

Checkpointing Application Preparation

A normal application that needs to be checkpointed does not need source-level modifications. FIGURE 5 shows how to checkpoint an application. The application source or object only needs to be relinked with the Condor checkpointing libraries to take advantage of the checkpointing and remote system calls. The Condor libraries contains an easy mechanism that helps to perform the relink operation by using the condor_compile command as follows:

```
condor_compile -condor_standalone command [options|files ...]
```

Where *command* is any of cc, f77, f90, or ld, and where [*options*|*files* ...] are the normal arguments used by the compiler and linker.

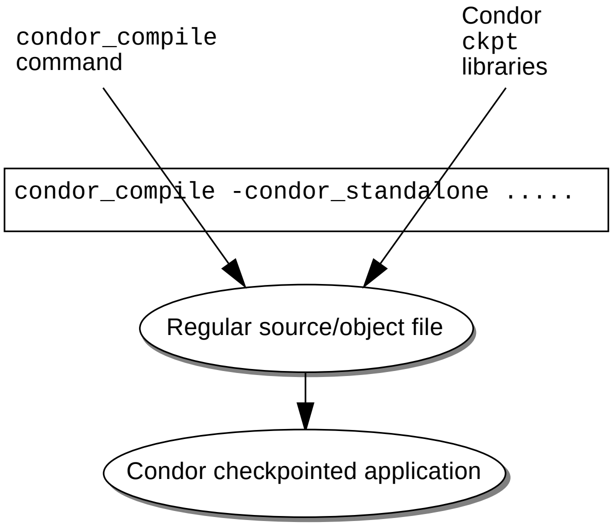

condor_compile
command
Condor
ckpt
libraries
condor_compile -condor_standalone .....
Regular source/object file
Condor checkpointed application

FIGURE 5 Checkpointing an Application

User-Level Checkpointing Deployment

The submission of a checkpoint job to a Sun GE environment is similar to the submission of a regular job, with the addition of the following options to the `qsub`(1) command:

- `-ckpt` *checkpoint_env_name*
- `-c [m|s|n|x]`

FIGURE 6 shows how a checkpointing job is submitted.

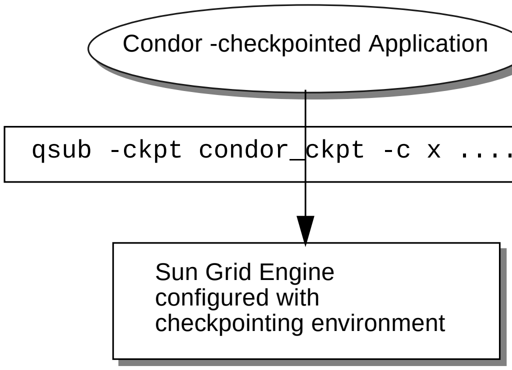

Condor -checkpointed Application
qsub -ckpt condor_ckpt -c x ....
Sun Grid Engine
configured with
checkpointing environment

FIGURE 6 Submitting a Checkpointing Application to the Sun GE Environment

In the Sun lab experiment, the `-c x` option was used because the job was to be checkpointed only when it was suspended. The Sun GE software provides other possibilities, and you should consult the `qsub`(1) man page to find out more about what behavior is desired for your specific application.

Migration of Checkpointing Jobs

The Sun GE software provides several ways to initiate the job migration capability. FIGURE 7 shows the framework of the migration feature. In the Sun lab experiment, the job suspension and the queue suspension to trigger the job migration were tested.

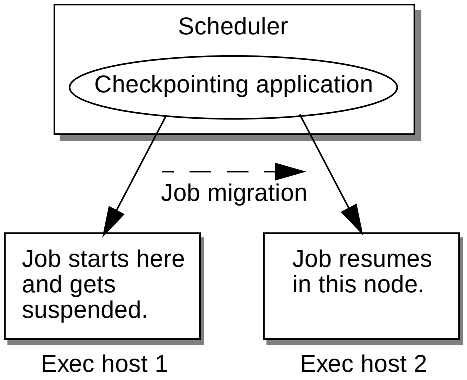

Scheduler
Checkpointing application
Job migration
Job starts here and gets suspended.
Job resumes in this node.
Exec host 1
Exec host 2

FIGURE 7 Migrating a Checkpointing Application

You can use the following procedure to apply job migration for a checkpointing application.

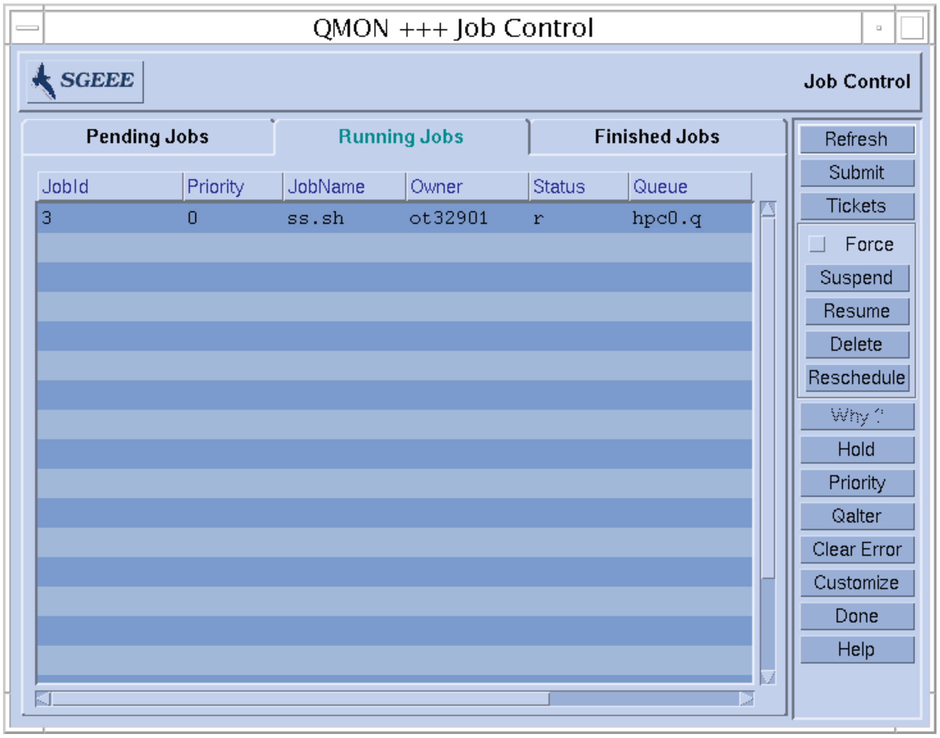
▼ To Migrate a Job Using the Sun GE Software

1. **Type the following** `qsub`**(1) command:**

```
qsub -ckpt condor_ckpt -c x ...
```

2. **Use the** `qmon` **graphical window to monitor the job execution on a particular queue.**

3. **Open** `qmon` **the Job Control window, and suspend the job.**



JobId	Priority	JobName	Owner	Status	Queue
3	0	ss.sh	ot32901	r	hpc0.q

FIGURE 8 Job Control Window

The job then shows up on the queue of a second executable host.

4. **Suspend the job on the second host.**

The job should be migrated to the queue of the first execution host and be successfully completed. The migration feature was also tested with the queue getting suspended, instead of the job. The job migration also completed successfully in this case.

Condor User-Level Checkpointing Limitations

The Condor user-level checkpointing libraries have some limitations on jobs that it can transparently checkpoint and migrate. The following list contains some of the limitations:

- Multiprocess jobs are not supported.

  This includes system calls such as `fork()`, `exec()`, and `system()`. Consequently, MPI programs are not supported.

- Interprocess communication is not supported.

  This includes pipes, semaphores, and shared memory.

- Network communication must be short.

  A job may make network connections using system calls, such as `socket()`, but a network connection left open for long periods will delay checkpointing and migration.

- Sending or receiving the `SIGUSR2` or `SIGTSTP` signals is not allowed.

  These signals are reserved by the Condor system. Sending or receiving all other signals is allowed.

- Alarms, timers, and sleep calls are not allowed.

  This includes system calls such as `alarm()`, `gettimer()`, and `sleep()`.

- Multiple kernel-level threads are not supported.

  However, multiple user-level threads are supported.

- Memory mapped files are not supported.

  This includes system calls such as `mmap()` and `munmap()`.

- File locks are allowed, but they are not retained between checkpoints.
- All files must be opened read-only or write-only.

  A file opened for both reading and writing will cause trouble if a job must be rolled back to an old checkpoint image. For compatibility reasons, a file opened for both reading and writing will result in a warning, but not an error.

- A fair amount of disk space must be available on the submitting machine for storing checkpoint images.

  A checkpoint image is approximately equal to the virtual memory consumed by a job while it runs. If disk space is short, a special checkpoint server can be designated for storing all of the checkpoint images in a pool.

HPC Administration and Usage Tips

This section gives simple recommendations to Sun HPC system users and administrators on various issues that relate to configuration and usage of the Sun HPC software installation. The following is a list that addresses the most frequently encountered issues:

- Do not mix different versions of the hostname syntax for the cluster nodes to prevent an HPC installation from successfully completing.

  Use the output of the hostname command in the `hpc_config` file. Make sure the same syntax is used in the `/etc/hpc_system` and the `/.rhosts` or `/etc/sunhpc_rhosts` files.

- Provide a superuser (`root`) readable and writeable directory for synchronization.

  Usually the `hpc_config` file is saved in this directory, and all of the SYNC files used by the install script are created in this directory. Check the `sync` directory for correct permissions before starting the installation.

- Change the permissions to 0600 and the ownership to `root` on the `/.rhosts` and `/etc/sunhpc_rhosts` authentication files.

  Make sure that the authentication files contain the hostname of the cluster nodes, including the host on which they reside.

- Refresh the resource database.

  The resource database sometimes gets out of sync and needs to be refreshed. This is demonstrated by wrong and unexpected output from the CRE commands. You should stop the CRE daemons, remove the `/var/rdb-*` files, and restart the CRE daemons.

- Try to avoid NFS-type installations.

  Use SMP-local or cluster-local installations only. The latter type has generated more clean installations than the NFS type, due to the wide variations of network configurations.

- Do not remove the `/tmp/CRE-ctblfile` file because it is needed by the CRE software.

  A job spawned by the Sun CRE is closely tied to a the `/tmp/CRE-ctblfile` file that lives as long as the CRE daemons are running. Most computer sites have scripts that regularly clean up the `/tmp` directory. There have been instances where long running jobs that take days to complete have failed due to the unexpected disappearance of the `/tmp/CRE-ctblfile` file.

- Use the `-t` *`scale_factor`* option with the `mprun`(1) command to increase the timeout period.

Jobs that spawn a large number of processes may, on rare occasions, fail with the following message:

```
mprun: tmrte_proc_spawn: select: Operation
timed out: Operation timed out
```

This is due to the default timeout value used by the Sun CRE to spawn all of the processes of the job.

- Configure a large `/tmp` swap partition because the MPI programs running on a particular node use shared memory files that are mapped to the `/tmp` area.

  TABLE 1 contains two examples of shared memory sizes with respect to the number of processes running on the same SMP:

TABLE 1 Shared Memory Sizes Per Processes

Processes per Job	Required Shared Memory
2	35 Mbytes
16	85 Mbytes

- Keep MPI network traffic separate from administrative and other network traffic to improve MPI application performance.

The above tips and recommendations frequently reappear on the support forums. See the Frequently Asked Questions at the following site:

`http://supportforum.sun.com/clustertools`

Read the *Sun HPC ClusterTools 4 User's Guide* and the *Sun HPC ClusterTools 4 Product Notes* at the following site:

`http://docs.sun.com/`

Use the Sun Cluster Support forum at:

`http://supportforum.sun.com/clustertools`

There are several forums at this site that users and experts use to discuss issues that pertain to the Sun HPC ClusterTools and the Sun Grid Engine products.

Appendix

This appendix contains a copy of the /etc/sudoers file. It includes the necessary changes to administer the Sun HPC ClusterTools 4 software.

```
-------------Start of /etc/sudoers file-------------------
...
snip
...
#
Host_Alias HPCHOSTS=<hostname>,<hostname>,...
#
User_Alias HPCUSERS=<username>,<username>,...
# Used for HPC CT 3.1
Cmnd_Alias HPCCMNDS=/opt/SUNWhpc/bin/*,/etc/init.d/sunhpc*,\
/opt/SUNWhpc/etc/*,\
/opt/SUNWhpc/etc/isa.h\
HPCUSERS HPCHOSTS=HPCCMNDS
/opt/SUNWhpc/etc/sparc*/*,\
/opt/SUNWhpc/etc/pfs/sparc*/*,\
/opt/SUNWhpc/bin/Install_Utilities/*
#
...
snip
...
-------------End of /etc/sudoers file-------------------
```

Acknowledgements

I would like to thank all of my colleagues from the many HPC-related groups for reviewing the original SUPerG white paper and offering their valuable feedback and suggestions.

References

This section contains the references used in this article.

- Sun HPC ClusterTools 4 documentation set at:

  `http://docs.sun.com/`
- The Sudo software at:

  `http://www.courtesan.com/sudo`
- The Platform software at:

  `http://www.platform.com`
- The Condor project at:

  `http://www.cs.wisc.edu/condor`
- The Sun Gridware software at:

  `http://www.sun.com/gridware`
- The Sun HPC ClusterTools 4 software support forum at:

  `http://supportforum.sun.com/clustertools`
- The Sun GE software at:

  `http://gridengine.sunsource.net`
- “HPC Best Practices” presented at the SUPerG conference in Paris, France, April 2000
- Code examples on the Sun GE software site:

  `http://gridengine.sunsource.net/project/gridengine/howto/condorckpt.html`

About the Author

Omar Hassaine is a senior HPC engineer with over twenty years experience in the computer industry. Omar worked on two consecutive high-end SPARC servers—including the Sun Enterprise™ 10000 server—as a project leader in the system software group at Cray Research Business Systems Division and Sun Microsystems, Inc., respectively. Before Omar joined Cray, he was a compiler engineer in the area of source code optimizers for super-compilers at Kuck & Associates. Omar has authored and given several technical presentations at various Sun sponsored events, and he helped develop and teach HPC administration and programming courses. Omar designed and developed an HPC diagnostics tool that is being deployed at large HPC sites.

Ordering Sun Documents

The SunDocs^{SM} program provides more than 250 manuals from Sun Microsystems, Inc. If you live in the United States, Canada, Europe, or Japan, you can purchase documentation sets or individual manuals through this program.

Accessing Sun Documentation Online

The `docs.sun.com` web site enables you to access Sun technical documentation online. You can browse the `docs.sun.com` archive or search for a specific book title or subject. The URL is `http://docs.sun.com/`

To reference Sun BluePrints OnLine articles, visit the Sun BluePrints OnLine Web site at: `http://www.sun.com/blueprints/online.html`