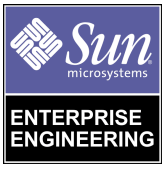
Sun HPC ClusterTools™ Software Best Practices

By Omar Hassaine - HES Engineering-HPC

Sun BluePrints™ OnLine - September 2000



http://www.sun.com/blueprints

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303 USA
650 960-1300 fax 650 969-9131

Part No.: 806-6202-10
Revision 01, September 2000

Please
Recycle

Sun HPC ClusterTools™ Software Best Practices

Abstract

This paper discusses the Best Practices for successfully configuring, installing and using the Sun HPC Clustertools™ software.

HPC Status in the Field

There are three major classes of HPC related problems at customer sites:

1. Install and configuration failures
2. Compile/link/runtime failures
3. Performance issues of user applications

In this paper, we will attempt to address the most important problems encountered in the field by providing few “Best Practices” for each of the major components of the HPC software such as:

1. The Sun™ Parallel File System(PFS)
2. Security and authentication issues
3. The Sun Message Passing Interface environment(MPI) and MPI Input/Output(MPI-IO)
4. The Sun Prism™ programming environment, and
5. The Scalable Scientific Subroutine Library(S3L)

Sun HPC ClusterTools Architecture

What follows is a brief list of the components that make up the Sun HPC ClusterTools software (refer to FIGURE 1).

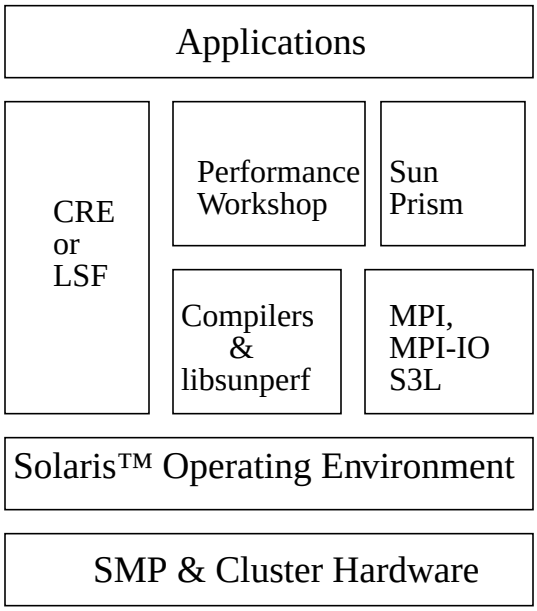


FIGURE 1 HPC Software Architecture

- Sun Cluster Runtime environment (CRE) or Load Sharing Facility (LSF) from Platform Computing Corp[1]
- Parallel File system
- Message Passing Interface
- MPI I/O, the extended MPI library for input/output. *Prism programming environment
- Scalable Scientific Subroutine library
- Cluster console manager
- Switch Management Agent

Further details about the Sun HPC ClusterTools software is found in the Sun HPC ClusterTools administration guide[1].

Sun HPC ClusterTools Software Best Practices

What follows is a set of Best Practices that apply to each of the components of the HPC ClusterTools software.

Sun HPC Installation and Configuration

The installation of the Sun HPC software has significantly improved in the 3.0 release of the Sun HPC ClusterTools. One of the main factors that has helped to improve the quality of software installation is the discontinuance of the FlexLM licensing from the Sun HPC ClusterTools 3.0 product. Another factor that has helped improve the software installation is the user friendly install GUI tool that was also introduced in the Sun HPC ClusterTools 3.0 release. Most of the install problems encountered now are related to authentication issues, NFS installations, and large cluster installations.

Authentication Issues

The Sun HPC ClusterTools software supports three different authentication settings: `none`, `krb5` and `des`. The `none` option and `CREATE_REMOTE_AUTH_FILE` option are related. If the latter is set to `YES`, then the file `/etc/sunhpc_rhosts` will be generated by the postinstall scripts with root readable permissions and the file will contain the names of the hosts in the cluster; otherwise, the `/.rhosts` file is used.

Kerberos v5 security and authentication technology is supported by Sun HPC ClusterTools 3.1 software on the Solaris Operating Environments 2.6, 7, and 8. Kerberos v4 will no longer be supported.

The Data Encryption Standard(DES) is also supported and can be used for secure RPC communications between HPC daemons.

The HPC administration guide[1] describes the use of Kerberos v5 and DES authentication systems and gives reference to other links for further details. A good practice when an authentication related problem occurs during the install phase is to revert to the `none` option. If the problem ceases, it suggests that the authentication mechanism is not properly configured.

NFS Installations

When selecting an NFS installation in the `hpc_config` file during the pre-installation phase, one has to be careful about choosing the paths of the installation and configuration directories. Based on our experiences, a good practice would be to refrain from using the `/net` string in directory paths and make sure that the directories used are accessible from any node in the cluster. We recommend using the option `cluster local` instead of `NFS` for performance reasons. The `cluster-local` option will enable every node to have its own local copy of the Sun HPC software so the HPC components are accessed and executed faster than if a remote NFS copy is invoked.

Large Cluster Installations

When installing the Sun HPC software on a large cluster, bear in mind that the Cluster Console Manager tools restrict the installation only to a maximum of 16 nodes at the same time. Also, when installing the HPC software with the Load Sharing Facility(LSF), it is recommended to select the `telnet` method instead of the `rsh` method in the `hpc_config` file as the utility to communicate between the nodes in the cluster during the installation stage. Currently, the installation scripts in Sun HPC ClusterTools 3.1 have a known bug and fail beyond 8 nodes when the `rsh` utility is used for an NFS installation. We have found that our postinstall scripts have been more stable with the `telnet` than with the `rsh` utility.

HPC Administration

The HPC administration should be minimal after a successful installation. There are however a few Best Practices that could help prevent future Sun HPC software failures:

- Before editing the `hpc.conf` file, the HPC daemons have to be stopped and will be restarted only after the `hpc.conf` has been updated and saved. Note that the `hpc.conf` resides in a different directory depending on which runtime environment is installed. For the LSF case, the file will live in the directory `$LSF_CONFDIR` that is initialized in `/etc/lsf.conf` file and in the CRE case, it is found in the directory `/opt/SUNWhpc/conf`.
- The `/etc/system` file needs to be updated to modify the shared memory related values when large programs are running out of shared memory space. A reboot of each node in the cluster is needed for the `/etc/system` file modifications to take effect. Setting `shmmax` to a maximum number is one of the

safest ways to proceed. The maximum value for `shmmax` depends on the Solaris Operating Environment version installed on the system. For example, the following values can be set in the `/etc/system` file.

For 32-bit Solaris:

```
set shmsys:shminfo_shmmax = 0xffffffff
```

and for 64-bit Solaris (Sol 7 & 8):

```
set shmsys:shminfo_shmmax = 0xffffffffffffffff
```

- The Sun HPC ClusterTools daemons rely on a resource data base that represents the state of the machine or the cluster and the jobs running in it. This database can get corrupted due to a power cycle or a hard panic. Also, the database can get out of sync when system administrators forget to update the database following a machine or a cluster reconfiguration. A good practice for HPC system administrators to use in these cases is to perform the following step on the master node:

```
sunhpc.cre_master reboot
```

and on all nodes execute the following:

```
sunhpc.cre_node reboot
```

The reboot option will purge the database of any transient data and restore only the configuration information. Please note that this option will not cause any system outage except for HPC applications which will need to be restarted after the steps described above are completed.

- Another good practice which is preventive and helpful to solve the corrupted and out of sync databases issue would be to run:

```
mpadmin -c dump >
            dumpfile.mycluster.<date>
```

after a successful installation and configuration, or after a cluster configuration change, to save the CRE data base information. The dump file can be real handy in the case of a hard crash. The CRE database will be restored to its original configuration by performing the following steps when the node is backed up.

Stop all HPC slave daemons on all nodes using:

```
/etc/sunhpc.cre_node stop
```

and on the master node type:

```
/etc/sunhpc.cre_master stop
```

Then, on each slave node:

```
rm /var/hpc/rdb*
```

Start all HPC daemons as follows:

```
/etc/sunhpc.cre_master start
/etc/sunhpc.cre_node start
```

Followed by the following command:

```
mpadmin -f
dumpfile.mycluster.<date>
```

Delete the HPC databases and start fresh when the above steps do not solve the problem.

Sun Parallel File System(PFS)

Sun PFS supports both Platform Computing's LSF and Sun's CRE. In either case, the system administrator needs to update the configuration file `hpc.conf` described earlier to configure Sun PFS. Sun PFS is highly recommended when high I/O is required. One way for applications to achieve the best performance using Sun PFS is to call the MPI I/O routines in the Sun MPI library.

Note that the Sun PFS file system cannot be mounted until all the Sun PFS daemons in the cluster are running. This is the reason it is advisable for system administrators to set the `mount at boot` option to `no` in the `/etc/vfstab` entry for any Sun PFS filesystem.

A good practice for setting the thread limits in the hpc.conf for Sun PFS would be to set it to 1 if the storage object is a single disk or a small storage array. The thread limit can be increased to make full use of the available I/O bandwidth of the disk subsystem when large I/O subsytems are used.

Another issue with Sun PFS is the collocation of the processes and the Sun PFS IO daemons(IODs). A good practice would be to collocate them if:

- It is a cluster of large multi-CPU machines
- There are fast storage devices
- A low performance cluster interconnect
- A small number of applications are competing for node resources

In the 2.x release of the Sun HPC software, Sun PFS was not tightly integrated with MPI so system administrators could configure the system to separate MPI network traffic from the Sun PFS network traffic. This possibility no longer exists in the 3.x release and there is no way we can separate the PFS network traffic from the MPI traffic. Sun PFS is now implemented using MPI and the intent for this new design was for Sun PFS to use the fastest interconnect namely the one using Sun Remote Shared Memory (RSM) Protocol over the Scalable Coherent Interface(SCI) network card.

A final note on Sun PFS is that in the 2.x release of the Sun HPC software, Sun PFS had separate commands from their UNIX® operating system counterparts. However, in the 3.x release, Sun PFS file systems appear in the UNIX system namespace, so users can use UNIX system commands such as cp, rm, ls and mv on files that live in the Sun PFS file system.

Sun Message Passing Interface

In this section, we attempt to describe a few Best Practices for system administrators and some basic tips for MPI programmers. The Performance Guide is a good resource document[1] that describes in detail how to analyze, debug and tune MPI programs.

Sun MPI Administration

The system administrator has the ability to set the default settings of the MPI options which control MPI communication behavior by editing the `MPIOptions` section in the `hpc.conf` file. The user also has the ability to override some but not all of the default MPI settings by using the MPI environment variables.

HPC system administrators need to keep in mind that Sun MPI uses shared memory for interprocess communication within one physical box or node. The Sun MPI shared memory protocol module uses a temporary storage file that resides in the `tmpfs` filesystem. The size of this temporary file is dependent upon the number of processes used and other MPI related parameters and can grow to the order of hundreds of MBytes. As a result, the sizing of the `/tmp` filesystem needs to take this temporary file into consideration to avoid MPI applications from running out of `/tmp` space.

Another good practice is to separate MPI network traffic from administrative and other network traffic. A customer case is worth mentioning in this article because the current HPC ClusterTools documentation does not describe the steps that the system administrator needs to perform to make MPI network traffic go through a specific port of the fast Ethernet (hme's) installed on the customer's system. The system traffic was going through the hme0 port because the original `hpc.conf` file has only one entry (`hme`) in the `Netifs` section. What needs to be done instead is to add another entry in the `hpc.conf` file for the hme1 port that will be used for the MPI network traffic as follows:

NAME	RANK	MTU	STRIPE	PROT	LAT	BW
`hme`	180	4096	0	tcp	20	150
`hme1`	179	4096	0	tcp	20	150

Notice that the rank of `hme1` is lower (higher priority) than `hme` so that MPI traffic will not go through the first port `hme0` which is implicitly specified by the `hme` entry.

When running MPI programs over Sun RSM, a good debugging practice would be to verify whether the Sun RSM daemon is running on all the nodes in the cluster. If a daemon is missing on one node, then the remaining ones need to be killed before restarting all of them on all nodes at the same time. Note that the RSM daemons need to be invoked as root using the following:

```
# /etc/init.d/sunhpc.hpc_rsmd start
```

Sun MPI user applications

A good debugging practice for MPI programs is to set the following environment variables to print out extra diagnostic information:

`#setenv MPI_PRINTENV 1` - Prints all the MPI environment variables settings and hpc.conf's MPI related parameters.

`#setenv MPI_SHOW_INTERFACES <1|2>` - Outputs the network interfaces used by MPI traffic.

`#setenv MPI_SHOW_ERRORS 1`-The MPI error handler prints out the error message and returns the error status.

Note – that the above environment variables need to be set only during the debugging phase of an MPI program because their use will incur a lower performance of the MPI application.

There are several MPI environment variables that affect directly or indirectly the performance of MPI programs. Their description and use is beyond the scope of this document and the interested reader should consult the performance guide document mentioned at the beginning of this section. However, I will cover the settings of the 3 most common environment variables that could potentially affect the performance of MPI programs. These environment variables need to be set before running the MPI application:

1. `MPI_SPIN` is the environment variable that sets the spin policy. Its default value is zero and causes the MPI processes not to spin aggressively. This setting delivers best performance when the load or run queue depth is at least as great as the number of processors. A value of **1** causes aggressive spinning and leads best performance if extra processors are available to handle system daemons and other background activities.

2. `MPI_PROCBIND` - When set to 1 binds each process to its own processor(default is **0**). The system administrator may allow or disable processor binding by setting the `pbind` parameter in the MPIOptions section of the `hpc.conf` file. It is not advisable to enable `MPI_PROCBIND` when there are multiple MPI jobs and/or multi-threaded jobs on a node. This has the potential to degrade performance because MPI processes will compete for the same processors and threads that could compete for the same processor. For Solaris Operating Environment 7 and beyond, one can bind processes to processors using `psrset`(1M).

3. `MPI_POLLALL` - When set to **1**, the default value, all connections are polled for receives(full polling). When set to **0**, only those connections where receives are posted are polled**.** Full polling helps drain buffers and so lessen the chance of deadlock or unsafe codes. Well written and tested codes should set `MPI_POLLALL` to **0** [for best performance].

Sun Scalable Subroutine Library-S3L

The Sun S3L library does not need any special settings from the system administrator since it is a library that is linked from the user's applications. In this section, it is worthwhile to point out the following user practices:

- Sun S3L calls functions in the Sun Performance Library(`libsunperf`) to perform various computations within each process. For best performance, make certain that the application uses the architecture specific version of `libsunperf`. This can be done by linking with `-xarch=v8plusa` for 32-bit executables and `-xarch=v9a` for 64-bit executables.
- Abstain from using the `LD_LIBRARY_PATH` environment variable as this can override link-time library selections and will make your application call a suboptimal library during runtime.
- A good S3L debugging practice is the use of the `S3L_SAFETY` environment variable. There are 3 debugging levels associated with this environment variable. The first level reports errors when more than one S3L function in a parallel program tries to use the same parallel array at the same time. The next higher level enables synchronization before and after each S3L call and reports any errors at each synchronization point. The third and highest level is useful for detailed debugging such as a check if multiple threads are trying to access the same S3L array. The higher the level the lower the performance due to increased checking into the S3L library code. The S3L interested readers should consult the Sun S3L Programming and Reference Guide[1].

Sun Prism Programming Environment

The Sun Prism programming environment also does not need any particular system settings by the system administrator. In this section, we will outline the following tips for HPC programmers to keep in mind when using the Prism environment:

- The Prism debugger works with both 32 and 64 bit binaries on the Solaris 7 Operating Environment and beyond. However, it cannot do performance analysis of 32-bit binaries in a 64-bit environment unless one includes the `-32` option in the Prism command line at invocation time.
- Loading code compiled with the `-xs` option may require long load times. This option is not required to run your code under Prism. If `-xs` was not used, it is required to keep the object(`.o`) files for Prism to find the debugging information. Note that the `-g` compile option is still needed to load programs in the Prism environment.
- Use only the Multiprocessing mode to load MPI programs using the `-n <#processes>` option when invoking Prism. Attempting to use the scalar mode (i.e. without the `-n` option) of the Prism programming environment to load an MPI program will abort the process and issue an error message.

Summary & Conclusion

We have attempted in this article to describe the most important Best Practices which hopefully will alleviate many of the problems that HPC system administrators and HPC programmers encounter when maintaining and using the Sun HPC ClusterTools software.

References

Sun HPC ClusterTools 3.1 documentation set: `http://docs.sun.com`

Author's Bio: Omar Hassaine

Omar Hassaine is a senior HPC engineer working in the High End Services group. Omar was previously a system software project leader for two consecutive high end SPARC server products.