Getting started with the CEES Grid

January, 2011

CEES HPTC Manager: Dennis Michael, dennis@stanford.edu, 723-2014, Mitchell Building room 415.

Please see our web site at http://cees.stanford.edu. Account requests, software requests and problem reports can be made via the forms on the drop down menu on 'HPTC Facilities'. If the problem is time critical, email me, phone me, or drop by my office.

The CEES Grid is accessed via the cluster head nodes described below. You will use your SUNetID and your SUNetID password to log in.

Using the clusters in the CEES Grid is relatively simple once a few basic commands are learned. The CEES Grid uses a batch processing system. You use a cluster in the CEES Grid by logging in to the head node for that cluster, compiling your code, and then submitting the code in a script to the batch system. The batch system schedules your job for execution on the appropriate compute nodes, and on completion of the job, writes the results in your directory or a location of your choice, and emails you with a completion notice. The batch system software we use is PBS/Torque with the Maui scheduler.

Examples of how to actually run a job are given in the 'example' section below. Your account has been set up with the environment you will need to get started.

Overview and Definitions

A *grid* is a collection of computing resources that perform tasks and appears to users as a large system that provides single points of access to distributed resources or clusters of resources.

A *cluster* is a collection of individual computers, a network connecting those computers, and software that enables a computer to share work among the other computers via the network.

The Clusters in the CEES Grid

The CEES Grid is composed of three resource clusters, each with a separate head node and separate compute nodes.

CEES Cluster

Operating System: Linux

Head node: cees-cluster.stanford.edu

- Hardware: dual Quad-core Nehalem (5520) cpus, 24 GB memory

Number of compute nodes in cluster: 160 organized in two partitions

- Compute node hardware: dual Quad-core Nehalem (5520) cpus, 24 GB memory

The CEES Cluster can be accessed by vested users only. See your PI.

CEES Opteron Cluster (Dev Cluster)
Operating System: CentOS Linux
Head node: cees-opteron.stanford.edu

- Hardware: Sun v40z with 4 dual-core Opteron CPUs and 16GB of memory

Compute nodes: 32 Sun V20z with 2 Opteron CPUs and 8 GB of memory

The Opteron Cluster has a 2 hour time limit for jobs. The Dev Cluster can be accessed by vested users only.

CEES Tool cluster (not really a cluster...)
Operating System: CentOS linux
Head nodes: cees-tool.stanford.edu and cees-tool-2.stanford.edu

- Each has dual Quad-core Nehalem (5520) CPUs with 48GB of memory

Software:

- GNU and open source software and compilers
- MATLAB (20 licenses), Comsol (2 licenses)

Usage:

- Primarily MATLAB and Comsol
- Programs that do not use MPI or SMP
- Users do not submit jobs via a batch system. Jobs are run from the command line.

Logging in to the Cluster head nodes

- You must use SSH or SFTP (on Windows, SecureCRT and SecureFX).
- You must log in from a Stanford host. If you wish to login from outside Stanford, you must first run the VPN software available at http://sussl.stanford.edu.

Disk Space
The Clusters share the same home directory partition and the data partitions.

- Home directory partition is 700GB, and users have a 10GB quota.
- The home directory partitions are RAID 5.
- The home directory partition is backed up (snapshot) daily
- The home directory is mounted read-only on compute nodes

In addition to the home directory, there are several directories in /data:

- The data partitions are a parallel file system and we have approximately 175 TB.
- File servers use RAID 5
- The data partitions are mounted in /data on the head nodes and the compute nodes
- Everyone has write access to the 20 TB in /data/cees. /data/cees is NOT BACKED UP!
- The other /data partitions are 'owned'. See your PI for access.
- There are no quotas in /data.
- Some /data partitions are backed up. See your PI.

Neither the home directory nor the /data partitions are designed for permanent storage. Please clean up your temporary files, and download your results as soon as possible. Remember that deleted files on partitions not backed up are not recoverable.

Please note the relative sizes of /home and the temp partitions. Use /home for source code; use the temp partitions for binaries, data sets and results.

Software Environment
Your account has been created with several default paths for system software. If you alter your .tcshrc, please retain the paths and commands. If you make a mistake and delete something you shouldn't in your .tcshrc, the default files are located in /etc/skel on the head nodes.

We have the GNU compilers and many open source programs available on the clusters. You can request new packages via the software request form on the web page.

If you have problems with the clusters, please report them via the web page.

Documentation
Linux

- CEES uses a Red Hat 'clone' called CentOS
- Red Hat docs are at http://redhat.com/docs/
- CentOS docs are at http://centos.org/
- Look in the install directory (usually /usr or /usr/local)
- Try the man command. Not all software has man pages.

Basic Commands and Examples
Examples and documents about PBS/Torque and the MPI packages can be found on the web.

All jobs are submitted to PBS/Torque via a job script which contains various definitions (such as a parallel environment), options, and the location of your binary executable program.

Some basic commands (see man pages for more options):

showq – show all running and queued jobs

checkjob <job #> - display information about a specific job. You can display only your own jobs.

pbsnodes: display details about the cluster nodes.

qsub – submits a job script to the batch system. After submission, the batch system will execute the job on the cluster and return the results to you. Examples of job scripts are below.

qstat - checks status of jobs.

qdel - deletes a job. Takes the job number as the argument.

Job Queues in the CEES Grid

The compute nodes in the Clusters are organized into queues. The Tool Cluster does not have any queues since the two servers are standalone and jobs are run from the command line.

First some comments about the network and the compute nodes. The compute nodes are organized into two network partitions corresponding to the two Infiniband network switches (named sw121 and sw39). Sw121 has 121 compute nodes connected to it, and sw39 has 39 compute nodes connected to it. There are a limited number of connections between the two switches, and MPI jobs that span the switches suffer severe performance degradation in both MPI and disk I/O. The batch system scheduler Maui will not allow jobs to span the switches.

Prof. Diffenbaugh's disk partitions are connected to sw39; all other disk partitions are connected to sw121. To ensure the best performance, Prof. Diffenbaugh's group should use the sw39 partition, and all other groups should use the sw121 partition. See the script examples below.

The batch system does not associate specific nodes with specific queues. The compute nodes are in a list and the batch system will use whatever node is free when a request comes in. Use the command 'pbsnodes' to see the list.

The CEES Cluster queue names:

default: contains all the unused nodes of the cluster. See below for important information about using large numbers of nodes. The default queue has a 2 hour time limit, and anyone on the cluster can submit jobs. Basically the default queue contains the unused nodes. Jobs submitted to the default queue have a lower queue priority than jobs submitted to the other queues.

The following queues are access controlled:

Q1: 'Mavko', unlimited run times.
Q10: 'Thomas', unlimited run times.
Q19: 'CEES', unlimited run times.
Q26a: 'ERE', unlimited run times.
Q26b: 'Dunham', unlimited run times.
Q2a: 'Beroza', unlimited run times.
Q2b: 'Harris', unlimited run times.
Q2c: 'Hilley', unlimited run times.

Q35: 'SEP', unlimited run times.
Q39: 'Diffenbaugh', unlimited run times.

EXAMPLE: ***a MPI version of 'hello world'***

Using your favorite editor, create a program file using the following code, and name the file 'hellompi.c'.

```
#include <stdio.h>
#include <mpi.h>
int main(int argc, char * argv[])
{
        int myid, numprocs;
        MPI_Init(&argc,&argv);
        MPI_Comm_size(MPI_COMM_WORLD, &numprocs);
        MPI_Comm_rank (MPI_COMM_WORLD,&myid);
        printf("Hello World! I am #%d of %d procs\n",myid,numprocs);
        MPI_Finalize();
}
```

Compile it using the command

```
% mpicc –o hellompi hellompi.c
```

Using your favorite editor, create a script file using the following code, and name the script 'testrun.sh'.

```
#!/bin/tcsh
#PBS -N TestJob
#PBS -l nodes=1:ppn=8
#PBS -q default
#PBS -V
#PBS -m e
#PBS -W x="PARTITION:sw121"
#PBS -M <YOUR SUNETID>@stanford.edu
#PBS -e /data/cees/<path to your directory>/test.err
#PBS -o /data/cees/<path to your directory>/test.out
#
cd $PBS_O_WORKDIR
#
mpirun hellompi >> OUT
```

Description:
line 1: shell to execute under
l**ine 2**: name of job. This is displayed in showq.
Line 3: nodes and cores. This line requests 8 cores on one node (note that each of our compute nodes has 8 cores, so this command requests a 'whole' node).

Line 4: what queue to run on. Make sure you have access. Here we are using the default queue. The default queue contains all the nodes in the cluster, and has a 2 hour time limit for jobs.
Line 5 and 6: Use these. See the manual for description.
Line 7: Important – you must use this line to specify the network partition. See above.
Line 8: Maui will email you at this address when the job is done
Line 9 and 10: path names for the error log and output log. Important – this cannot be your home directory. The /home partition is mounted read only on the compute nodes, and trying to use it will cause this script to fail.
Line 11: comment line
Line 12: 'cd' to the working directory (the directory you are submitting the job from)
Line 13: comment line
Line 14: Command to execute, plus any options. This line will vary according to the application. Most programs will not have to specify the number of slots and the hostfile to mpirun. However, if you receive an error saying something about 'found only 1 processor', try using the following format:

mpirun -np [# of procs] -machinefile $PBS_NODEFILE [program and input] >> OUT

Note that the # of procs must agree with the '-l' line.

Example: mpirun -np 8 -machinefile $PBS_NODEFILE hellompi >> OUT

You can look at the /data/cees/dennis direcory for many other examples. That is where I test out software and commands, and you are welcome to look and/or use my scripts.

Please email dennis@stanford.edu if you have any difficulties or problems.