Iowa High Performance Computing Summer School 2010

Getting Online

Welcome to the second annual Iowa High Performance Computing Summer School, May 25-26, 2010. In addition to the presentation on the board, I have compiled these notes as a reference to help you get online on Moffett, the 4536-core SiCortex 5832 at the Rosen Center for Advanced Computing at Purdue University, and on the research clusters here at the University of Iowa.

1 Logging onto and Running Parallel Programs on Moffett

The directions in this section will get you logged onto and running on Moffett.

1. Moffett is a 4536-core SiCortex 5832 located at the Rosen Center for Advanced Computing at Purdue University. You will be set up with a guest account on this machine during your participation in this course. Your guest account *username* and *password* will be given to you on paper. Detailed instructions on the use of Moffett can be found online at
   `http://www.rcac.purdue.edu/userinfo/resources/moffett/newuser.cfm`

2. To get connected to Moffett, be sure you have Xwindows running on your machine and pull up an xterm window. Connect to moffett using ssh,
   `ssh -X` *username*`@moffett.rcac.purdue.edu`
   and enter your *password* at the prompt. This will put you into your home directory on Moffett. Note that the `-X` enables Xforwarding so that you can pull up an Xwindow of an application running on the remote machine (Moffett) on the monitor of your local machine.

3. We are going to copy and example parallel code, `HYDRO`, to Moffett using `scp`. Follow the instructions below:

   (a) Create a directory named `hydro` in your home directory on Moffett
   `mkdir hydro`

   (b) Go to the IHPC 2010 website at `http://www.physics.uiowa.edu/~ghowes/teach/ihpc09/index.html` and follow the Examples link. Download the tar file of `HYDRO`, `hyd100524.tar`, to a directory on your local machine.

   (c) Open a new xterm window on your machine and navigate to the directory in which you just put `hyd100524.tar`.

   (d) Now, we will copy this tar file over to Moffett using `scp`
   `scp hyd100524.tar` *username*`@moffett.rcac.purdue.edu:~/hydro/`
   and enter your *password* at the prompt.

   (e) In the window on Moffett, go into the `hydro` directory and unpack the tar file
   `tar -xvf hyd100524.tar`

4. Compiling the parallel code `HYDRO`:

   (a) The tar file for `HYDRO` contains a `Makefile` that will allows for easy compilation of the code on different platforms (different computers). This code is written in Fortran90 using MPI for parallelization.

   (b) Compile the code by typing
   `make`
   This will produce an executable `hydro.e`

5. Running parallel programs on Moffett

   (a) Unlike many parallel computers which use Portable Batch System, or PBS, for job scheduling, Moffett uses the Simple Linux Utility for Resource Management, or SLURM. Although the syntax of the commands differs between these two systems, in general the operation is similar. Here we will give you a few of the basics on running parallel codes using SLURM on Moffett. Much more detail can be found online at `http://www.rcac.purdue.edu/userinfo/resources/moffett/newuser.cfm`

   (b) The examples below will use `HYDRO` with the sample input file `sample1.in`. The code requires the first argument after the executable to be the input file, thus the command will be
   `hydro.e sample1.in`

   (c) We can choose to run either interactively or in batch mode. Interactive runs generally run immediately (if resources are available), whereas running in batch mode puts the job into a queue to be run when resources become available. As we write our first parallel codes today, we will generally run in interactive mode, since we want the results right away (and since we have resources reserved for this course). Typically, when running your codes at a national supercomputing center, you will almost exclusively run in batch mode.

   (d) INTERACTIVE MODE: To run interactively, the following syntax applies
   `srun -p <partition> -n <tasks> <executable> [args]`
   Here, we select the partition on Moffett set up for this class `scx-event`. The regular partition on Moffett is `scx-comp`, to which you may also submit jobs. The number of tasks is the number of MPI processes, or equivalently the number of cores you choose to use. In the case of running `HYDRO` on 16 cores, we would use
   `srun -p scx-event -n 16 ./hydro.e sample1.in`
   In this case, because you are running interactively, the output of `HYDRO` for this run will be sent to standard output (the screen).

   (e) To check the queue, use the command
   `squeue`

   (f) To cancel a job that has been submitted or kill a job that is running, use
   `scancel <jobid>`

   (g) BATCH MODE: The usual method for running on a shared cluster or at a national supercomputing center is to run in batch mode, submitting your jobs using a script. The scripts for SLURM are different from those using PBS, so here we will provide an example for using SLURM on Moffett. Here is a shell script named `sample1_moffett.sh` for running the `sample1.in` run with `HYDRO` using 16 cores

```
#!/bin/sh
#SBATCH -n 16
#SBATCH -t 00:10:00
#SBATCH -o sample1.log
#SBATCH -e sample1.err
#SBATCH -J sample1
#SBATCH -p scx-event
echo Running hydro on 16 processors
srun ./hydro.e sample1.in
echo hydro run complete
```

   Each of the options on the lines above specifies a different aspect of the run:
   `-n` specifies the number of cores,
   `-t` specifies the time limit in HH:MM:SS format,
   `-o` specifies the name of the file to send the standard output,
   `-e` specifies the name of the file to send the error output,
   `-J` specifies the name of the submitted job,

`-p` specifies the partition on which to submit the job.
The echo lines above simply write to the log file a few useful comments, but are not necessary.
To submit the job, use
`sbatch sample1_moffett.sh`
You may then check to see that the job is in the queue or running using `squeue`.

2 Logging onto and Running Parallel Programs on University of Iowa Research Clusters

Research Services of Information Technology Services and the Institute for Clinical and Translational Research at the University of Iowa have two small clusters that are generally open to use by University of Iowa researchers:
rs-001.its.uiowa.edu 44-core, 32-bit cluster
rs-003.its.uiowa.edu 64-core, 64-bit cluster
Below we give instructions for logging onto and running on these machines, again using `HYDRO` as an example. The instructions will be given for running on the 64-bit cluster rs-003. Running on the 32-bit is essentially the same.

1. Connect to rs-003 using
   `ssh -X` *username*`@rs-003.its.uiowa.edu`
   and enter the password recently supplied to you over e-mail at the prompt.
2. To use the 64-bit cluster, go into the `home64` directory.
3. We are going to copy and example parallel code, `HYDRO`, to rs-003 using `scp`. Follow the instructions below:
   (a) Create a directory named `hydro` in this home directory
   `mkdir hydro`
   (b) Go to the IHPC 2010 website at `http://www.physics.uiowa.edu/~ghowes/teach/ihpc09/index.html` and follow the Examples link. Download the tar file of `HYDRO`, `hyd100524.tar`, to a directory on your local machine.
   (c) Open a new xterm window on your machine and navigate to the directory in which you just put `hyd100524.tar`.
   (d) Now, we will copy this tar file over to rs-003 using `scp`
   `scp hyd100524.tar` *username*`@rs-003.its.uiowa.edu:~/hom64/hydro/`
   and enter your *password* at the prompt.
   (e) In the window on rs-003, go into the `hydro` directory and unpack the tar file
   `tar -xvf hyd100524.tar`
4. Compiling the parallel code `HYDRO`:
   (a) The tar file for `HYDRO` contains a `Makefile` that will allows for easy compilation of the code on different platforms (different computers). This code is written in Fortran90 using MPI for parallelization.
   (b) First, open up the `Makefile` using emacs running in the background
   `emacs Makefile &`
   Edit the Makefile by changing the SYSTEM options variable
   `SYSTEM=uiowa`
   (c) Compile the code by typing
   `make`
   This will produce an executable `hydro.e`

5. Running parallel programs on rs-003

   (a) The 64-bit cluster rs-003 uses a combination of TORQUE and Maui for scheduling and resource allocation. This is much more similar to the systems in place at most national supercomputing centers than SLURM used on Moffett.

   (b) BATCH MODE: Only batch mode is allowed for submission of parallel jobs on rs-003. Here is a shell script named `sample1_rs-003.sh` for running the `sample1.in` run with `HYDRO` using 16 cores

```
#PBS -S /bin/tcsh
#PBS -V
#PBS -N sample1
#PBS -o sample1.log
#PBS -j oe
#PBS -l walltime=00:50:00
#PBS -l nodes=4:ppn=4
#PBS -m abe
#PBS -M username@uiowa.edu
echo "Job begin:"`date`
cd $PBS_O_WORKDIR
mpirun -np 16 ./hydro.e sample1.in
echo "Job end:"`date`
```

   The echo lines above simply write to the log file a few useful comments, but are not necessary. Be sure to put in your *username* for the e-mail address.

   To submit the job, use

```
qsub sample1_rs-003.sh
```

   You may then check to see that the job is in the queue or running using `showq`.