Iowa High Performance Computing Summer School 2015

Getting Online

Welcome to the sixth annual Iowa High Performance Computing Summer School, June 1–3, 2015. I have compiled these notes as a reference to help you get online on Neon, the 4216-core shared cluster here at the University of Iowa, with 11 Nvidia Kepler K20 GPU cards and 38 Xeon Phi accelerator cards.

1 ITS Computer Training Room Workstations: Connecting to Neon

For this course, we will be using the Windows PCs in the ITS Computer Training Room, 2523 UCC. Here I outline how to get set up on the local workstation for easy use during this course and how to connect remotely to Neon.

1. If the computer at your workstation displays a Mac username and password input, tap the "Scroll Lock" key twice rapidly to toggle to the Windows PC.

2. Log onto the workstation using your HawkID and password.

3. First, you want to launch the xwindows server Xming so that you can pull up windows on your local terminal screen from applications running remotely on Neon. To do this, start XLaunch from the Windows Start Menu at the lower left corner of the Screen. Note that XLaunch may be under Xming in the All Programs folder.

   (a) Choose display setting "Multiple Windows."

   (b) Choose "Start no client" to just start Xming, allowing us to start local clients later.

   (c) For additional parameters, leave as is and just select Next.

4. Start SecureCRT 7.3 under All Programs from the Start Menu.

   (a) Starting SecureCRT should pull up a Quick Connect Window. You should choose the following options:
   Protocol: SSH2
   Hostname: neon.hpc.uiowa.edu
   Port: 22
   Firewall: None
   Username: *Your HawkID*
   Authentication: All boxes checked

   (b) You may get a "New Host Key" dialog box, asking you to verify the host key. Click Accept & Save.

   (c) At first when the Enter Secure Shell Password window comes up, select Cancel.

   (d) Next, right click on the tab showing neon.hpc.uiowa.edu and select "Session Options . . ."

   (e) Under the PortForwarding item, choose Remote/X11.

   (f) Under X11 forwarding, check the "Forward X11 Packets" box and select OK.

   (g) Right click on the neon.hpc.uiowa.edu tab again and select Reconnect.

   (h) In the Enter Secure Shell Password, enter your HawkID password.

   (i) Next, in the SecureCRT terminal window, will appear the Duo two-factor authentication request. Option (1), the Duo Push to your device, is usually the easiest option if you have the Duo app installed.

(j) Once you have authenticated the Duo push request on your secondary device, you will be logged onto Neon remotely, and the Neon prompt should appear, for example
`[ghowes@neon-login-0-1 ~]$`
At this point, you may begin issuing Unix commands on Neon.

(k) To log out, simply type `exit` at the prompt and you will be disconnected.

2 Running Parallel Programs on Neon

The directions in this section will get you running parallel programs on Neon.

1. Neon is a 4216-core shared cluster computer on the University of Iowa campus. The compute nodes consist of the following configurations:
   (A) 188 compute nodes of:
   2.6 GHz 16 Core Processors (Standard Nodes)
   64 GB Memory
   (B) 59 compute nodes of:
   2.6 GHz 16 Core Processors (Mid-Memory Nodes)
   256 GB Memory
   (C) 11 compute nodes of:
   2.9 GHz 24 Core Processors (High-Memory Nodes)
   512 GB Memory
   In addition, there are 38 Xeon Phi 5110P Accelerator Cards and 11 Nvidia Kepler K20 Accelerator Cards installed on certain nodes in the cluster.

2. Online documentation for Neon can be found at
   `https://wiki.uiowa.edu/display/hpcdocs/Neon+Cluster+Documentation`

3. Module Set Up: The first time, and only the first time, that we log onto Neon, we need to set up our `.bashrc` file so that the software modules that we need are loaded automatically.

   (a) To view the modules that are already loaded, use the command `module list`, and to see available modules, use `module avail`.

   (b) To add the necessary lines to the `.bashrc` file, in your home directory, open the `.bashrc` file using `emacs` in the background using
   `emacs .bashrc &`
   A window will open showing the contents of your `.bashrc` file. Note that if the font size in emacs is too small to read comfortably, you may change the font size using `CTRL-X CTRL-+`
   At the end of the file, after the line
   `# User specific aliases and functions`
   add the following two lines
   ```
   export SYSTEM=neon
   module load openmpi
   ```
   Next hit `CTRL-x CTRL-s` to save the changes, and `CTRL-x CTRL-c` to quit `emacs`.

   (c) Now you may exit your login shell and follow the previous instruction to log in via `ssh` again. Once you have logged on again, you may use `module list` to verify that the necessary modules are loaded.

4. We are going to download the example parallel code, `HYDRO`, to Neon. Follow the instructions below:

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

(b) Navigate into the hydro directory,
`cd hydro`

(c) Type the following command to download the tar archive of the `HYDRO` code
`curl -o hyd150529.tar http://newton.physics.uiowa.edu/~ghowes/teach/ihpc15/codes/hyd150529.tar`

(d) Note that the `HYDRO` code is also available at the IHPC 2015 website at
`http://www.physics.uiowa.edu/~ghowes/teach/ihpc15/index.html`
under the Examples link.

(e) Inside the `hydro` directory, unpack the tar file
`tar -xvf hyd150529.tar`

5. Compiling the parallel code `HYDRO`:

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

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

6. Running parallel programs on Neon

(a) Neon uses the Sun Grid Engine (SGE) as a scheduler for parallel jobs. More information can be found online at
`https://wiki.uiowa.edu/display/hpcdocs/Basic+Job+Submission`

(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 will be running in batch mode, the preferable method for submitting jobs to shared computing resources. Submitting a job in batch mode puts the job into a queue to be run when resources become available.

(d) 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.

i. An example shell script for running on Neon is included in the tar archive, `sample1_neon.sh`, for submitting a batch job to SGE to run `HYDRO` (for the input file `sample1.in`) follows:

```
#!/bin/sh
# Job Submission script

#$ -q IHPC
#$ -pe 16cpn 16
#$ -l h_rt=00:10:00
#$ -N sample1
#$ -o sample1.log
#$ -j y
#$ -V
#$ -cwd

echo "Job begin:"`date`
echo "Run sample1 on Neon, 16 proc"
mpirun -n 16 hydro.e sample1.in
echo "Job end:"`date`
```

ii. Each of the options on the lines above specifies a different aspect of the run:
`-q IHPC` specifies the IHPC training queue for the job (other options are `UI` or `all.q`)
`-l h_rt=00:10:00` specifies the time limit in HH:MM:SS format
`-pe 16cpn 16` requests 16 cores using 16 cores per node
`-N sample1` specifies the name name of the job
`-o sample1.log` specifies the name of the file to send the standard output
`-j y` merges the stdout and stderr output,
`-V` imports environmental variables to the parallel SGE environment
`-cwd` places the output files in the current working directory.
The echo lines above simply write to the log file a few useful comments, but are not necessary.

iii. To submit the job, use
`qsub sample1_neon.sh`

iv. You may then check to see that the job is in the queue or running using `qstat -u` *username*. This will produce output that looks like
```
job-ID prior name user state submit/start at queue slots ja-task-ID
-----------------------------------------------------------------------------------------------------------------
736050 0.50944 sample1 ghowes r 05/31/2012 21:59:58 IHPC@compute-6-177.local 16
```
The `job-ID` in this example is `736050`.

v. To delete a job, submit a `qdel` command followed by the job-ID
`qdel 736050`

3 Compiling Your Own Parallel Code

To compile an MPI code on Neon

1. FORTRAN 90: If your code is name `program.f90`, you can compile using
`mpif90 -o program.e program.f90`
where the option `-o program.e` names the resulting executable `program.e` (the default behavior is to name the resulting executable `a.out`).

2. C: If your code is name `program.c`, you can compile using
`mpicc -o program.e program.c`
where the option `-o program.e` names the resulting executable `program.e` (the default behavior is to name the resulting executable `a.out`).

3. Note that on Neon, you must be sure that the following modules are loaded
`openmpi`
You should have this set up in your `.bashrc` file, but if not, you can load the modules yourself by issuing the `module load` commands yourself, for example, `module load openmpi`.