High Performance Computing and the UI Neon Cluster

Kate Cowles

Department of Statistics and Actuarial Science
The University of Iowa

ISIB
June 25, 2014

What to do when a computing problem is too big, or computing would take toolong, for a single computer to handle

- parallel computing
- using graphical processing units for mathematical calculations
- distributed and grid computing

Parallel computing

- many calculations carried out at once
- multi-core or multi-processor computers: more than one processing unit in a single machine
- computer clusters: multiple computers networked together to be able to work simultaneously on same computing problem

Challenges of parallel computing

- communication between processors more time intensive than calculation
  - more of an issue in distributed-memory systems than shared memory
- so problems that can be decomposed into small pieces that can execute independently are most amenable to parallel solutions

General programming with graphical processing units

- high end graphics cards such as those used in gaming computers have many cores for rendering graphics (up to several hundred)
- recently languages and programming toolkits have been developed to enable the use of the cores in graphics cards for mathematical computation
- challenges are the same as parallel processing on a cluster
  - has to be possible to divide the whole computing job (or parts of it) into individual small tasks that can be executed independently in parallel
  - slow transfer of data and results between CPU (regular processor) and GPU (graphics processor)

What is the Neon cluster?

`https://wiki.uiowa.edu/display/hpcdocs/Neon+Overview+and+Quick+Start+Guide`

- a High Performance Computing System
- runs CentOS 6.3, a Linux operating system based on Red Hat Linux.
- 2 login nodes
- 157 compute nodes
- 85 TB of shared storage

Compute nodes

- 131 64GB Nodes, 2.6GHz 16 Core (Standard Nodes)
- 17 256GB Nodes, 2.6GHz 16 Core (Mid-Memory Nodes)*
- 9 512GB Nodes, 2.9GHz 24 Core (High-Memory Nodes)
- 29 Xeon Phi 5110P Accelerator Cards*
- 3 Nvidia Kepler K20 Accelerator Cards*

* Statistics node is mid-memory with both Xeon Phi and Nvidia Kepler GPU

Nvidia Kepler K20 GPU

`http://www.nvidia.com/object/tesla-servers.html`

- 2496 CUDA cores
- peak double-precision floating point performance: 1.17 Tflops (trillion floating point operations per second)
- contrast with the NVIDIA Quadro 6000 GPUs in four machines in 346
  - 448 CUDA cores
  - 0.51 Tflops

Neon nodes owned by Biostatistics and Statistics departments

- Biostats department owns 2
- Statistics department owns 2 nodes
  - one with both Nvidia Kepler and a Xeon Phi
  - one with no accelerators but high memory

Accessing Neon

- accessed through SSH
- from a Linux machine on campus
  `ssh hawkid@neon.hpc.uiowa.edu`
- from a Linux machine off campus
  `ssh -p 40 hawkid@neon.hpc.uiowa.edu`
- use CSG version of NoMachine NX client to access Neon from a Windows or Mac machine
  `http://www.divms.uiowa.edu/help/windows/nomachine/`
  `http://www.divms.uiowa.edu/help/macosx/nomachine/`

Login nodes

- prompt to enter Hawkid password
- do not run jobs on login nodes
  - See “Performing Computations" section of Quick-Start guide
- either
  - log in to a compute node and work there
  - (preferred) submit your job(s) to queues so they will run on appropriate compute node(s)

Transferring files to Neon

- each account has 1 TB of disk space in home directory
- use `scp` or `sftp` (with port number if you are doing so from off-campus computer)
- example: I’m logged in to the DIVMS network. I want to copy file called "drive.R" from my current directory on the DIVMS network to my home directory on Neon

```
scp drive.R neon.hpc.uiowa.edu:
```

- example 2: I’m logged onto my Linux computer at home and want to copy a file of the same name to Neon

```
scp -P 40 drive.R neon.hpc.uiowa.edu:
```

Queues on Neon

`https://wiki.uiowa.edu/display/hpcdocs/Queue+Usage+and+Policies`

- investor queues – physical machines assigned to the departments or groups that purchased them
- UI – centrally funded; default queue
- all.q – all nodes (all job slots); available to anyone with an account

Batch scheduler on Neon – Sun Grid Engine (SGE)

- has commands for navigating compute nodes and for submitting, controlling, and monitoring jobs submitted to them

Logging in to the Statistics dept node, “LT"

`https://wiki.uiowa.edu/display/hpcdocs/Qlogin+for+Interactive+Sessions`

- `qlogin -q LT` or `qlogin -q LT -l kepler`
- to run R, must load its *environment module*
  `module load R`
  `https://wiki.uiowa.edu/display/hpcdocs/Environment+Modules`
- then can run R interactively or in batch mode

Interactive login, continued

- use `qlogin` rarely
  - ties up entire node for you; no one else can log on or submit jobs to it
  - to restrict to using just some of the 16 slots (4 in the example)
    `qlogin -q LT -pe smp 4`
  - be sure to “exit" to leave node
- can make sure your interactive session has ended by using `qstat` on login node to list all of your jobs

```
qstat -u $USER
```

Submitting jobs to run on compute nodes

- do this from a login node to direct job to run on a particular queue
- need executable script (and possibly additional R program(s))
- example

```
qsub -cwd -q LT cmd.job
```

  - `-cwd` – use current working directory
  - `cmd.job` is name of script to be executed

Monitoring and controlling jobs: `qstat`

- gives listing of all jobs that have been submitted on cluster and haven't finished
- includes
  - job-ID – what you need to use to delete job
  - user
  - state
    - `qw` – enqueued and waiting
    - `qr` – enqueued and running
  - slots – number used by job

More on `qstat`

- `qstat -u $USER`
- `qstat -q LT`

Deleting jobs: `qdel`

- use `qstat` to get job id
- then `qdel < job id >`

Where to get more information

- Patrick Breheny's tutorial
  `http://myweb.uiowa.edu/pbreheny/neon/`
- Patrick's follow-up live tutorial 3/24
- ITS documentation for Neon
  `https://wiki.uiowa.edu/display/hpcdocs/Neon+Overview+and+Quick+Start+Guide`
- online tutorials for Sun Grid Engine
- Matt Bognar's web page
- in near future, similar page on using the NVIDIA Kepler GPU