High Performance Computing

Jeff, Jun 9, 2017

But first...

- Go over a few key points from the Software Carpentry lesson on the unix shell.
- Shell scripts:
  - Are text files.
  - Allow you to store commands and re-use later
  - Accept arguments: **“$1” “$2” “$@”**
  - And allow you to automate via loops:
    - **for** b **in** blah
    - **do**
    - ***<commands..>***
    - **done**
- Part 6 of the lessons.

What is high performance computing?

- Usually it is what people mean when they use the term supercomputer or cluster
-
- Compute power varies greatly, but the organizational scheme is usually the same
-
- Physically the computer is “somewhere else” ie in the UDC in our case and you must access it via the network
-
- Hardware/software -wise there are Head nodes / compute nodes
- Job scheduler.

Usually you need to:

- Use scp or Filezilla (sftp) to transfer your code and data to the remote system.
- Use ssh to log in to the head node.
- Here you will have access to the CLI of the shell on the head node to organize files/data and compile code.
- You also use the shell to submit jobs to the queue to run on the compute nodes. The scheduler takes care of managing the queue your code runs when its at the front of the line and the resources are available.

Matlab bandpass filtering code example

- In matlab on my laptop: 244MB of data in 1 file = ~130s
- Using multiple cores:
- In matlab on my laptop (with parfor): 2 files = 130s (with parallel pool already up) = 200s (if you need to start a new parallel pool)
- 1.3-2X faster as expected
- but that's all the cores my laptop's got!
-
- So what now?
-

Seek a higher power...

- Alder address: alder.arc.ubc.ca
- What is it?
- 9x 32-core 128GB+ machine grouped together.
- Compute jobs are submitted to a queue and executed when resources are available.
-
- When is it useful?
- When you have something that is:
  - Established, ie. no more debugging, changes
  - No user interaction required
  - You need to do a lot (justifies the overhead of moving data and code and learning to use the cluster)

Alder example,

- Log in to alder and do the hello world and filter test examples with the matlab compiler.
-
- At this point, I am still working out the parallel computing toolbox with the System Admin, so we don't quite have that working yet.
- So I can't demo the amazing speed up we would get if using all 32-cores at once, but even a single core there is faster than my laptop.
-

So how to proceed with your application?

- Develop your code locally
- Test on a small amount of data/files which you will scale up
- Use scp or filezilla to transfer your stuff to alder
- Use the matlab compiler to compile your code!
- Test on the same small batch by submitting your job to the queue
- If that works as expected, go for a bigger batch of data!