

Informal introduction to supercomputing

Charlie Shobe

August 28, 2016

Contents

1	Introduction	1
1.1	Acknowledgements	1
1.2	Structure of Beach	2
1.3	Why use a supercomputer?	2
2	Cautionary notes	3
2.1	Don't run jobs on the head node	3
2.2	Don't use "sudo"	3

1 Introduction

This is a short cheat sheet to help get readers up to speed on how to drive the CU-CSDMS HPCC, more informally called "Beach." It assumes only very basic knowledge of terminal commands and the command line. Probably the things described here will be more difficult on a Windows OS than a Mac/Linux OS.

1.1 Acknowledgements

My limited understanding of supercomputing is thanks to Mariela Perignon, Mark Piper, Eric Hutton (all CSDMS), and Lauren Wheeler (U. New Mexico). All code and workflows shown here were developed in conjunction with one or more of them.

1.2 Structure of Beach

Beach is a collection of many “nodes,” or individual computers that are networked together. Most nodes are “compute nodes,” meaning that they are the nodes responsible for running your slow fancy model. The whole show is run by the “head node,” which is the node you are automatically working in when you log in to Beach. Anything you formally ask Beach to run is called a “job.” The head node is responsible for taking user job submissions and parsing them out to the different compute nodes to be run. To control the flow of jobs from users through the head node to different compute nodes and back again, Beach uses a job management system called Torque. Because of that, driving Beach involves a combination of Linux command line commands (when you’re talking directly to the computer) and specialized Torque commands (when you’re talking to the job management system).

1.3 Why use a supercomputer?

Some supercomputers are blazing fast. Beach is not blazing fast. Each node is about as fast as a standard laptop. However, there are still major advantages to working on a supercomputer instead of your own machine:

1. Run multiple jobs at once

If you need to run your model many times, as most of us do, you can submit many jobs to Beach concurrently such that they will run at the same time. This means you’re no longer harnessing the power of one laptop, but ten or twenty laptops simultaneously. This has enabled me to do hugely expensive parameter space analysis in hours or days, as opposed to months.

2. Keep your computer free for development and debugging

Once a model is stable and simply needs to be run many times to generate data, those runs don’t need supervision but still take up a large amount of memory and processing power. Outsourcing runs that don’t need to be closely watched to a remote supercomputer frees up your personal computing resources for further development work even while your stable models are running.

3. It’s a form of backup

While keeping your models and driver files on a remote supercomputer should certainly not be the only form of backup you have, it’s handy to know that you can always retrieve those files (subject to admin restrictions of course). This

means that if you break your model on your computer or lose an essential file, it's easy to retrieve the stable version you stored on the supercomputer.

2 Cautionary notes

2.1 Don't run jobs on the head node

2.2 Don't use “sudo”