R: High-performance computing on Great LakesStatistics 506

High-performance computing

Even with the use of parallel processing and optimizing our code as much as possible, we will eventually reach the limit of what can be done on personal computers. High-performance Computing is a term to describe carrying out computing tasks on supercomputers or, more commonly, a cluster of machines.

From an end-user perspective, the use of a HPC environment is typically not much different than any other remote machine, for example VirtualSites. However, behind the scenes, a HPC environment often consists of many computers with a shared pool of resources, with hardware that vastly exceeds what's available on a personal machine.

For example, Great Lakes, which is UM's cluster (which we discuss below), has a pool of "Large Memory" machines with 1.5TB of memory (that's 1500 GB, whereas personal machines typically have 8-32 GB).

We will be covering using Great Lakes, but the general workflow will work for most other HPC environments you interact with, even if the details shift.

Great Lakes

Great Lakes is UM's general-usage cluster (there are a few more specific use-case clusters, such as ARMIS2 for HIPAA data). Great Lakes can be interacted with either via the command line, by a web-interface, or through a virtual desktop. These notes will cover the latter two, and thus will be mostly a series of links. In lecture, we will walk through the usage.

Information about Great Lakes can be found at https://arc.umich.edu/greatlakes/.

To access the cluster, visit https://greatlakes.arc-ts.umich.edu. Note that this link can only be accessed on Campus, or if connected through the VPN.

On that page, the "Home Directory" contains a web-interface to your files available on the cluster. You can upload/download directly from there, or use RStudio to connect via Github while on the server.

We'll cover "Jobs" below.

The "Clusters" tab allows access to the cluster by command line; if you're going this route, I'd recommend SSH'ing in directly instead.

Finally, "Interactive Apps" launches a remote desktop. The "Basic Desktop" gives the most functionality (it's a complete linux desktop environment), but you'll probably find the "RStudio" application sufficient.

Allocations

Great Lakes is not free - you must have an allocation to bill to use it. You can check which allocations you have access to at https://portal.arc.umich.edu/projects. You should have an allocation called "stats506f23-class", and you may have others. If you are playing around with Great Lakes, I recommend using the "stats506f23-class" while you can as it doesn't cost anything, whereas if you have another allocation through a lab or something, it may incur charges.

When using Great Lakes, you'll need to specify which allocation to use.

Basic Desktop

When launching a Basic Desktop, you may need to request special software. See https://arc.umich.edu/greatlakes/software/ for the full list; for our purposes, both SAS and Stata fall into this category.

By default, no software is loaded. You'll need to use the `module` command to load software. Launch "Terminal Emulator". The following commands may be useful:

```
# Show all statistics programs:
module keyword statistics
# Search for a specific program:
module spider sas
# Load software
module load RStudio
module load stata-se
# See all loaded software
module list
# Unload a module (e.g. if you need to change version)
module unload stata-se
```

More documentation can be found at https://arc.umich.edu/greatlakes/software/lmod/.

To launch software, you'll need to enter a command as well.

Software	Command
RStudio	`rstudio`
Stata-SE (terminal)	`stata-se`
Stata-SE (gui)	`xstata-se`
Stata-MP (terminal)	`stata-mp`
Stata-MP (GUI)	`xstata-mp`
SAS	`sas`

Submitting Jobs

Working interactively has limitations:

1. We need to actually be connected,
2. There's processing overhead that is not needed (e.g. a running version of RStudio on a desktop)

Instead, we can use Slurm to submit "Jobs" which are basically just scripts, telling Great Lakes to run our code and return the results.

Jobs can be submitted at https://greatlakes.arc-ts.umich.edu/pun/sys/dashboard/apps/show/myjobs

The simulation repository I created for these notes is at https://github.com/josherrickson/506_simulation, and specifically the scripts https://github.com/josherrickson/506_simulation/blob/main/simulation.sh and https://github.com/josherrickson/506_simulation/blob/main/simulation-futures.sh.

These are Slurm scripts; they tell Great Lakes what resources to use and then the final lines are the actual code to run. There's a full guide to Slurm on the Great Lakes website: https://arc.umich.edu/greatlakes/slurm-user-guide/.

You can view your active (and historical) jobs at https://greatlakes.arc-ts.umich.edu/pun/sys/dashboard/activejobs.

Other useful links

If you are comfortable working at the terminal/command line, the following "Cheat Sheet" may be useful for you: https://arc.umich.edu/wp-content/uploads/sites/4/2020/05/Great-Lakes-Cheat-Sheet.pdf.

A full "user-guide" to Great Lakes is https://arc.umich.edu/greatlakes/user-guide/