STATS 810 class 12: Introduction to parallel statistical computing on greatlakes

Edward L. Ionides

Outline

Requirements

We follow the greatlakes user guide for getting started with the command line. As preliminaries, you need:

- A Slurm account. You should already have a primary account, `stats_dept1`, and a smaller backup account for if you exhaust your resourses, `stats_dept2`.
- A greatlakes cluster login account. If you have not yet filled in the form at `https://its.umich.edu/advanced-research-computing/high-performance-computing/great-lakes/getting-started` then do so.
- A umich internet address. Use the umich VPN if you are not on campus.

Connecting to greatlakes with macOS, Linux or Windows

Use the umich VPN if you are not on campus.

1. Open a Terminal window and type

   ```
   ssh uniqname@greatlakes.arc-ts.umich.edu
   ```

   where `uniqname` is your uniqname.
2. Login with your Kerberos level-1 password, and Duo two-factor authentication.

This creates a remote terminal shell on greatlakes.

Connecting to greatlakes with a browser

Use the umich VPN if you are not on campus.

1. Point your browser to `https://greatlakes.arc-ts.umich.edu`
2. Choose the menu option: Clusters → Great Lakes Shell Access

This creates a remote terminal shell on greatlakes within your browser.

Moving files on and off greatlakes: scp

On Mac or Linux, you can use `scp` which has similar syntax to `cp`.
To copy `myfile` on your laptop to a subdirectory `mydir` of your home directory on greatlakes:

```
scp myfile uniqname@greatlakes-xfer.arc-ts.umich.edu:mydir
```

To copy an entire directory, use the `-r` flag for recursive copy:

```
scp -r mydir uniqname@greatlakes-xfer.arc-ts.umich.edu:
```

These commands can also be reversed to copy files from greatlakes to your machine. The following copies `mydir` back to the current working directory:

```
scp -r uniqname@greatlakes-xfer.arc-ts.umich.edu:mydir .
```

You will need to authenticate via Duo to complete the file transfer. On Mac or Windows, FileZilla provides a file system user interface.

Cluster batch workflow

1. You create a batch script and submit it as a job
2. Your job is scheduled, and it enters the queue
3. When its turn arrives, your job will execute the batch script
4. Your script has access to all applications and data
5. When your script completes, anything it sent to standard output and error are saved in files stored in your submission directory
6. You can ask that email be sent to you when your jobs starts, ends, or fails
7. You can check on the status of your job at any time, or delete it if it's not doing what you want
8. A short time after your job completes, it disappears

Useful batch commands

Submit a job

```
sbatch sample.sbat
```

Query job status

```
squeue -j jobid
squeue -u uniqname
```

Delete a job

```
scancel jobid
```

Check a job script and estimate its start time

```
sbatch --test-only sample.sbat
```

More Slurm commands to try

`sacct -u user`	show recent job history
`seff jobid`	show cpu utilization for jobid
`my_accounts`	show all billing accounts on which you can run jobs

All Stats PhD students should have access to the `stats_dept1` and `stats_dept2` accounts.

R modules on greatlakes

Sotware on greatlakes is packaged in modules which must be loaded

```
module load R
```

Other versions of R are available:

```
module avail R
```

We see that R4.4.0 is currently the default. For simple multicore computing, the default R module is appropriate. Other versions of R have been built and tested in other parallel environments, for example the `Rmpi` module runs R with mpi.
R packages can be installed using `install.packages` within R, run at a terminal on the login node. Your home directory files (and therefore these packages) are accessible on all nodes of the cluster.

Set up test for foreach

- The `gl` subdirectory of the `810f25` git repository has a file `test.sbat` which submits a batch job running the parallel `foreach` test in `test.R`.
- You can transfer the files from your laptop via `scp`, or by copy-paste. Or simply clone the class git repository into your greatlakes account,

```
git clone https://github.com/ionides/810f25.git
```

- Inspect the text file `test.sbat`, for example by

```
more test.sbat
```

- One thing that needs changing is to set your email address for alerts about jobs beginning and ending. To make these edits on greatlakes, you need a text editor. Options include

```
vi test.sbat
nano test.sbat
```

- It is useful to have basic familiarity with these editors, and you can ask the internet for help.

Comparing results

- You are now ready to run a batch job

```
sbatch test.sbat
```

- From inspecting the code in `test.R`, we see that the results are saved in `test.csv`
- Compare the run times with the results from running this code on your laptop, as done in homework 9.
- Also, try running the code in `test2.R` by

```
sbatch test2.sbat
```

What do you learn from comparing the outputs in `test2.csv` with `test.csv` on greatlakes and your laptop?

Other ways to run R on greatlakes

- It is sometimes useful to start an interactive session on greatlakes, particularly for debugging. This is done from the terminal as follows:

```
module load R
srun --nodes=1 --account=stats_dept1 --ntasks-per-node=8 \
  --pty /bin/bash
```

- You can then run R in the terminal by typing

```
R
```

- This R session will have access to the cores you have requested.
- Here, we require `nodes=1` unless we use `Rmpi` since `library(doParallel)` alone cannot work with cores across different machines.
- You can also run web-based Rstudio. However, your task here is to run batch jobs, which remain the basic tool for intensive statistical computing.

Acknowledgments

- This lesson is prepared for STATS 810, Fall 2025.
- It builds on previous versions of STATS 810 and notes by Charles Antonelli and John Thiels.
- Compiled on November 17, 2025.