Greatlakes using Python with Pypomp for STATS 531 W26

Edward Ionides

Do I need to use greatlakes for STATS 531

- **No, it is optional, but recommended.** A modern laptop is sufficient to complete a reasonable STATS 531 final project.
- Long time series, or high-dimensional models, will be hard to investigate using particle filter methods without access to a reasonably powerful machine. Profile likelihood calculations can be computationally demanding, since they require many maximizations.
- JAX, and therefore Pypomp, is designed to take advantage of GPU capabilities. The same code can be tested on a Mac or greatlakes CPU, and then run on a greatlakes GPU.
- If your project becomes computationally ambitious, GPU computing is recommended. You can scale back the task to fit on a laptop, if necessary.

Requirements to log in to greatlakes

We follow the Great Lakes online documentation. You need:

- **A Slurm account**. Everybody in this class is a member of the account `stats531w26s001_class`. Graduate students in the Applied Statistics and Data Science masters programs, or Statistics PhD program, also have a primary departmental account, `stats_dept1`.
- **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

1. Open a Terminal window (recall that, on a Mac, this can be done using Control-Spacebar and typing Terminal) 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. At time of writing, greatlakes has not replaced Duo with Okta.

This creates a remote terminal shell on greatlakes.

Connecting to greatlakes with a browser

- Use the umich VPN if you are not on campus.
- Point your browser to https://greatlakes.arc-ts.umich.edu
- Choose the menu option: Clusters → Great Lakes Shell Access. This creates a remote terminal shell on greatlakes within your browser.

Synchronizing files using git

- If you have a project on GitHub, clone the git repository to greatlakes and push your work back to greatlakes when you are done.
- This workflow keeps your project synchronized between greatlakes, GitHub, and your laptop.
- It is recommended to clone the class git repository into your greatlakes account,

  ```
  git clone https://github.com/ionides/531w26.git
  ```

  If you have an ssh key set up in your GitHub account (recommended) you can use

  ```
  git clone git@github.com:ionides/531w26
  ```

Moving files on and off greatlakes using scp

scp which has similar syntax to the terminal command 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. To copy mydir back to your machine:

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

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

A workflow for working with batch jobs

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

Command	Description
`sacct -u user`	show recent job history
`seff jobid`	show cpu utilization for jobid
`my_accounts`	list accounts you have permission to use

Python modules on greatlakes

Software on greatlakes is packaged in modules which must be loaded

```
module load python
```

Other versions of Python are available:

```
module avail python
```

The default, marked `(D)`, is currently 3.13.2.

A virtual environment for JAX on CPU

- The use of virtual environments is always recommended.
- Here, it is useful to have a Python version set up for CPU and a version set up for GPU. The CPU version simply uses `pip install jax`

```
python -m venv ~/.venv-cpu
source ~/.venv-cpu/bin/activate
pip install --upgrade pip
pip install jax
pip install pypomp
```

- Running

  ```
  pip freeze
  ```

  reveals this also installs `jaxlib`, `numpy`, `scipy` and some other packages.

A virtual environment for JAX on GPU

- To set up a new virtual environment, first shut down the current one:

```
deactivate
```

- Then, build a new one, using jax[cuda12]

```
python -m venv ~/.venv-gpu
source ~/.venv-gpu/bin/activate
pip install --upgrade pip
pip install jax[cuda12]
pip install pypomp
```

- Once you have loaded the module and your virtual environment, you can start a Python session in a terminal on the login node just by typing `python`.
- The login node is appropriate for setting up Python environments and other administrative tasks.
- You are not supposed to do heavy multi-core computing on the login node. Short tests are okay.
- Your Python jobs run via `sbatch` can load the pre-setup virtual environment.

An interactive GPU sesson on greatlakes

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

```
srun --partition=gpu --account=stats531w26s001_class \
  --gpus=v100:1 --cpus-per-gpu=1 --pty /bin/bash
```

You can then run Python in the terminal as usual, for example the `531w26` git repository has a file `greatlakes/test.py` which runs some demonstration JAX code.

```
module load python
source ~/.venv-gpu/bin/activate
cd ~/531w26/greatlakes
python test.py
```

When you want to leave the interactive session,

```
exit
```

A test for batch computing

- `531w16/greatlakes/test-cpu.sbat` contains a SLURM script to submit a batch CPU job running the test in `test.py`.
- Before running this script, please change the email address to your own, so you receive the notifications about the job starting and ending. If you're not sure how to edit a text file on Linux, see below for more details.
- Then, run the job by

```
sbatch test-cpu.sbat
```

- The output is in the `.out` file corresponding the job. It may be interesting to study these run times, and compare greatlakes with your laptop, though the main goal here is just to practice running the code.
- Many tutorials on command line Linux are online, e.g. https://ubuntu.com/tutorials/command-line-for-beginners.

Editing text files on greatlakes

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

```
more test-cpu.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. It is convenient to use a text editor that runs in a terminal. Options include:

```
vi test-cpu.sbat
nano test-cpu.sbat
```

- It is useful to gain some familiarity with these editors.
- This Python session will have access to the cores you have requested. Here, we require `nodes=1` since `multiprocessing` alone cannot work with cores spread across different machines.
- You can also run web-based Jupyter. However, batch jobs remain the basic tool for intensive statistical computing.

Acknowledgments

- This lesson builds on the Great Lakes User Guide, an introduction by Charles Antonelli and John Thiels, and notes from STATS 810.
- Compiled on March 24, 2026.