Make an account on Carbonate (if you haven't yet!)

From the IU Knowledge Base "Get additional IU computing accounts"

1. Go to "Create More Accounts"
2. Click "Create Computing Account".
3. Select "Carbonate" from the list

You will be asked to supply your country of citizenship, primary, secondary and tertiary disciplines and subdisciplines and agree to some terms.

Anne Kavalerchik
Indiana University Bloomington
Departments of Sociology and Complex Systems
Email correspondence to: akavaler@iu.edu

Workshop in Methods (WIM), Fall 2021

Introduction to Supercomputing for Social Scientists

Goals

- Introduce high-performance computing (HPC)
- Introduce the command line interface (CLI)
- Practice using the CLI to access HPC
- Understand best practices for CLI and HPC

Part 1

Why use supercomputing?

Benefits of HPC

Increased processing and memory capacity

- Ability to work with large datasets and/or more complicated analyses
- Ability to keep your personal machine free by running jobs remotely

Data storage

- Data sets like the Web of Science and Microsoft Academic Graph are stored on HPC clusters
- Can use Slate to store data securely **and** share with collaborators

Additionally: HPC at IU is constantly maintained by the Research Technologies team, and comes pre-loaded with the most common libraries

Benefits of understanding the command line

- Using the CLI can seem intimidating. It shouldn't be!
- Understanding the CLI helps you understand your machine better
- Using the CLI allows for greater control and management of your operating system – like installing and removing software
- Advanced computational tools, like HPC, require some understanding of the CLI to access

Part 2

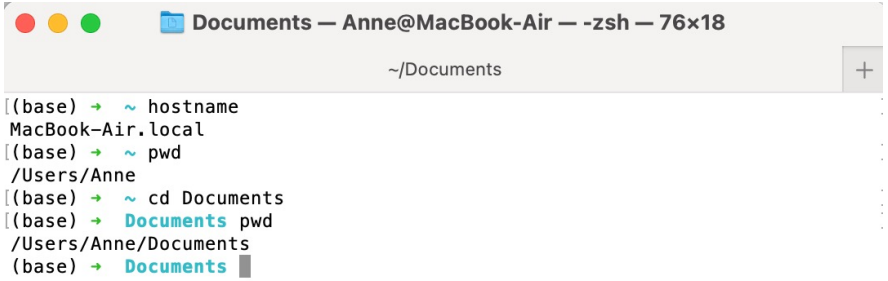
The Command Line

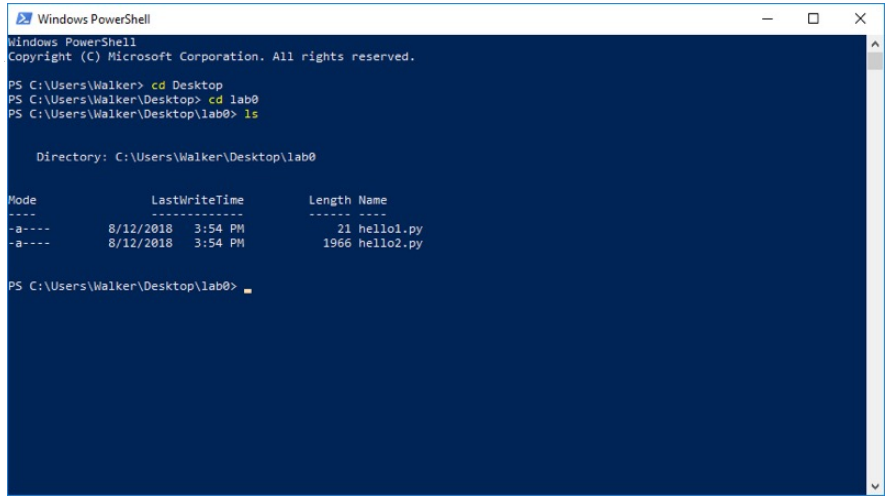
The Command Line?

A Command Line Interface (CLI) is a way to interact with a computer program through lines of text

Read-**E**val-**P**rint **L**oop

Related words: Bash, Shell, Terminal, Unix, Linux





Accessing the Shell

- **macOS**: Launch Terminal (Command + space “Terminal”, or going into /Applications/Utilities)
- **Linux:** Launch Terminal (usually via Ctrl + Alt + t)
- **Windows:**
  - Installed Shell program (launch through Windows + R)
    - PowerShell (pre-installed on Windows 10) – so all IU computers should have it
    - Git Bash
    - Windows Subsystem
    - Cygwin
  - Once installed, launch from Windows start menu or by pressing Windows + R and typing application name

Part 3

Accessing HPC Resources

Research Desktop (RED)

The RED provides access to HPC recourses through a **graphical user interface (GUI)**. Users can launch RED from their local machines through the software application ThinLinc.

UITS resources on using the RED:

- Download, install, and configure ThinLinc Client to use RED at IU
- About RED at IU
- RED usage policies and interface features

ssh: Secure Shell

ssh is a **network communication protocol** that allows computers to communicate with one another.

The process of connecting to an IU a supercomputing cluster through Terminal is similar to that of accessing other IU resources like email, Canvas, etc. You will use your **username**, **password**, and **2-factor authentication (Duo)**

We are going to use the following command (also on the next slide)

```
ssh [your username]@carbonate.uits.iu.edu
```

Connecting with ssh

We use the following command:

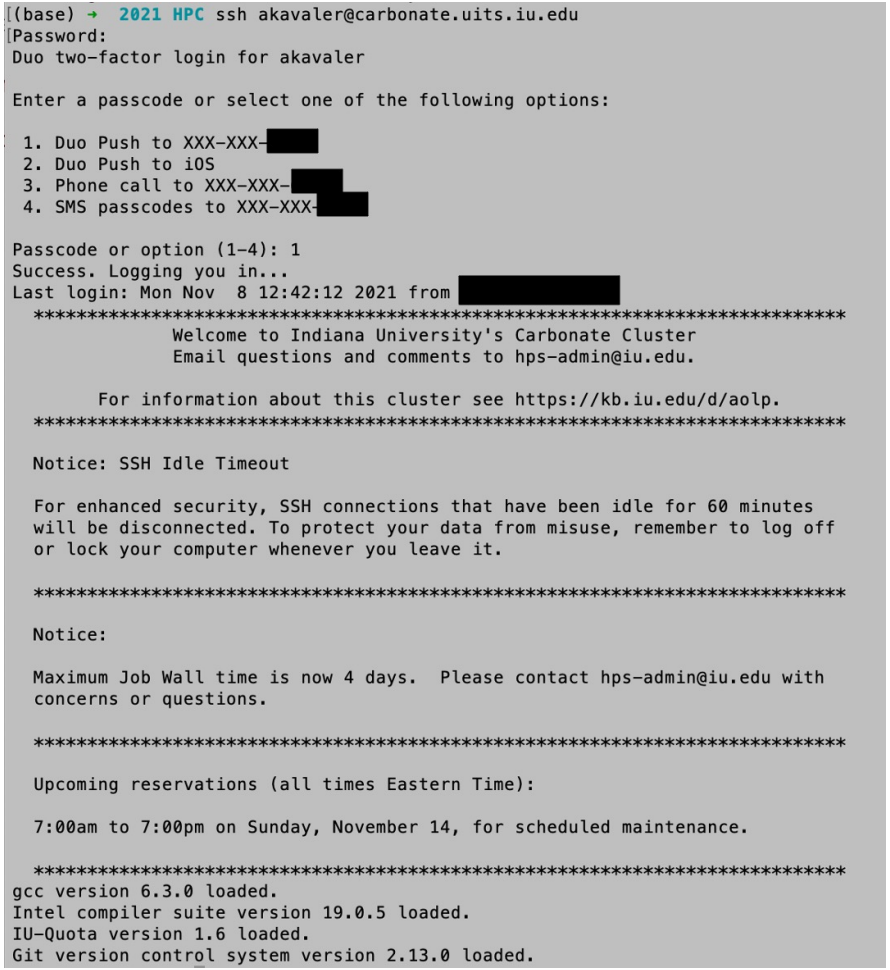
```
ssh [username]@carbonate.uits.iu.edu
```

Then it will prompt you to enter your password – this is what you use to access email, Canvas, etc. **Note that it will not show asterisks or spaces as you are entering it**.

Then it will prompt you for 2 Factor Authentication

Successful completion should look like the screenshot on the right:

To logout, type `logout`



Navigating files and folders

`pwd` print working directory
`ls` list directory
`cd [directory]` change directory
`cd ..` go “up” in the directory path
`cd ~/` go to home directory
`cat [file]` print contents of file
`cp [source] [dest.]` copies file/direc.
`mv [source] [dest.]` moves file/direc.

Writing files and folders

`touch [file]` create new file
`nano [file]` open file in nano
`mkdir [directory]` make new directory
`rm [file]` remove file

To remove a directory (see note):
`rmdir [empty directory]`
`rmdir -r [nonempty directory]`

Note:

1. **Be careful with** `rmdir -r`**!**
2. There are text editing options other than `nano`
3. Press `ctrl + c` to terminate any shell process

CLI Practice

1. Navigate to Documents (hint: under home directory)
2. Create a new directory called “hpc_practice”
3. Create a text file inside this directory called “practice.txt”
4. Write to and save the text file
5. Check the contents of the text file

CLI Practice answers

1. Navigate to Documents
   - `cd ~/Documents` or `cd /N/u/akavaler/Carbonate/Documents`
2. Create a new directory called “hpc_practice”
   - `mkdir hpc_practice`
3. Create a text file called “practice.txt” inside this directory
   - `touch practice.txt`
4. Write to and save the text file
   - `nano practice.txt`
5. Check the contents of the text file
   - `cat practice.txt`

Part 4

Transferring files

sftp: Secure file transfer protocol

Similar to ssh:

- Requires IU credentials and 2-factor authentication
- Many of the same commands for navigating

Intended for **transferring files** from local to remote machine

- Commands outside of that purpose don't work – e.g., `nano` doesn't work

Connecting with sftp

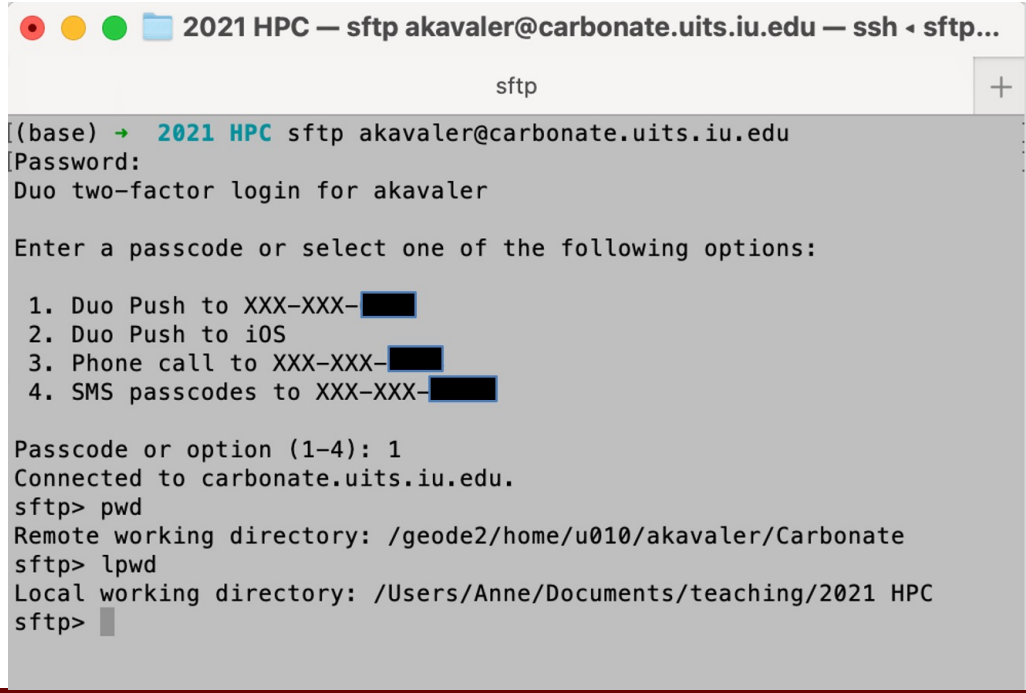
`sftp [username]@carbonate.uits.iu.edu`

Letter “l” (lowercase L) in front of command denotes whether it is intended for local or remote (e.g., `ls` vs `lls`, `pwd` vs `lpwd`)

`get` and `put` commands do the work of moving files back and forth

- `get` copies **from** remote
- `put` copies **to** remote

To log out, type `bye`



SFTP Practice

1. Navigate to the remote directory where practice.txt is saved
2. Copy practice.txt to your local machine
3. Open and write something to practice.txt, whether through nano or another text editor
4. Copy practice.txt to Slate storage space (/N/slate/[username])
5. Print contents of the file now stored in Slate (use SSH)

SFTP Practice answers

1. Navigate to the remote directory where practice.txt is saved
   - `cd Documents/hpc_practice`
2. Copy practice.txt to your local machine
   - `get practice.txt`
3. Open and write something to practice.txt
   - `nano practice.txt` or `open practice.txt`
4. Copy practice.txt to Slate storage space
   - `put practice.txt /N/slate/[username]/practice.txt`
5. Print contents of the file now stored in Slate (use SSH)
   - `cat /N/slate/[username]/practice.txt`

Other considerations

Moving lots of files can take a long time

- For moving something just once, combining and/or compressing with `tar` or `zip`
- For maintaining a backup: `rsync`. I recommend using rsyncinator to generate commands with desired specifications

Best practices

Check your code!

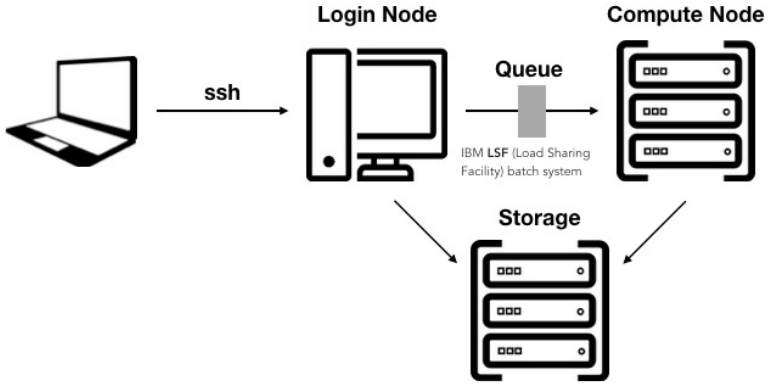
- Be careful with path names in your code
- Commands `cat [filename]`, `head [filename]`, `tail [filename]` allow you to print files in the terminal
- `nano[filename]` allows you to edit files
- Notice: Code in Shell environment less easy to read – keep your code neat!
- Name file and folders concisely and descriptively

Part 5

Submitting jobs

What is an HPC cluster?

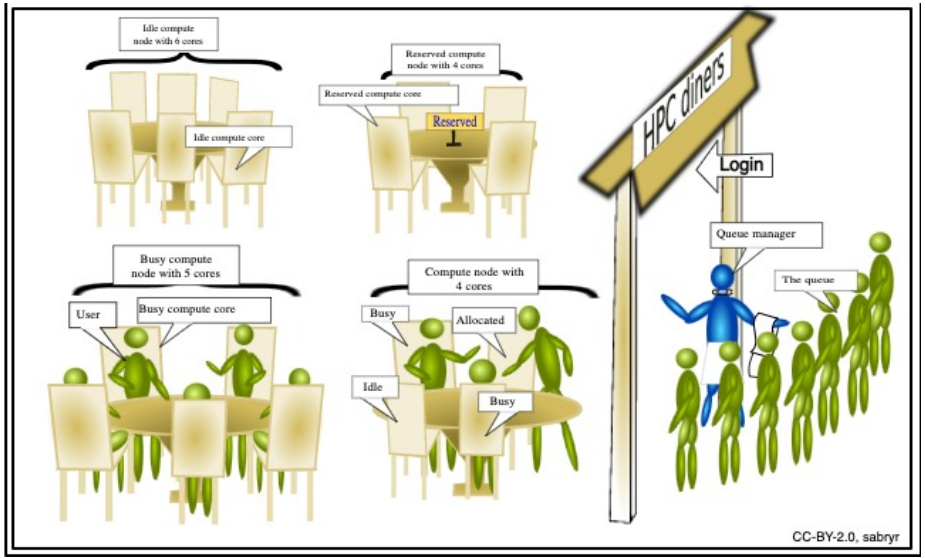
- An HPC cluster is a collection of many nodes, where each node is a computer pared down to the basic components of **CPU** (processing power**), RAM** (memory), and **disk space**
- Nodes are composed of **cores** (aka processors)
- A program that a user wants to run on the HPC is called a **job**
- HPC clusters use a method of **job scheduling** to allocate jobs to nodes



Job queues/Partitions

- An HPC job scheduler determines which jobs get to which nodes based on the resources that each job needs and the resources that are available
- Resources include available memory and nodes
- IU HPC uses a software called SLURM for this process

Illustration from https://epcced.github.io/hpc-intro/13-scheduler/index.html



SLURM: Simple Linux Utility for Resource Management

SLURM is a job scheduler

Scheduling a job with SLURM requires

1. Creating a SLURM script that tells the HPC cluster the script you want to execute and the resources you want to use

2. Executing the SLURM script

IU Knowledge Base: [Use SLURM to submit and manage jobs on high performance computing systems]

Basic SLURM Commands

- `squeue`: monitor job status information
- `scancel [job name]`: Terminate a queued or running job
- `sbatch [job name]`: submit a job (batch)
- `srun [job name]`: submit a job (interactive)
- `sinfo`: View node and partition info
- `salloc`: get a Slurm job allocation

Type `squeue` to see what jobs are running in real-time!

Batch jobs vs interactive jobs

Batch: `sbatch` submits a job to the partition (queue) for later execution – so you usually have to wait. It will be handled by the SLURM job scheduler – you can close your terminal, etc.

- Batch jobs also allow for **job arrays**, submitting and managing collections of similar jobs nearly simultaneously (documentation says milliseconds!)

Interactive: `srun` submits a job for real time execution. If you disconnect, they will terminate. Benefit: not waiting in the job queue. Good for troubleshooting

Installing & configuring modules

On HPC, software is bundled into “modules”. You specify which module you want in your Slurm script.

In Terminal:

- `module avail`: tells you which modules are available
- `module list`: tells you which modules you have currently pre-loaded

This is important if you need a specific version of a software, e.g. Python 2 vs Python 3

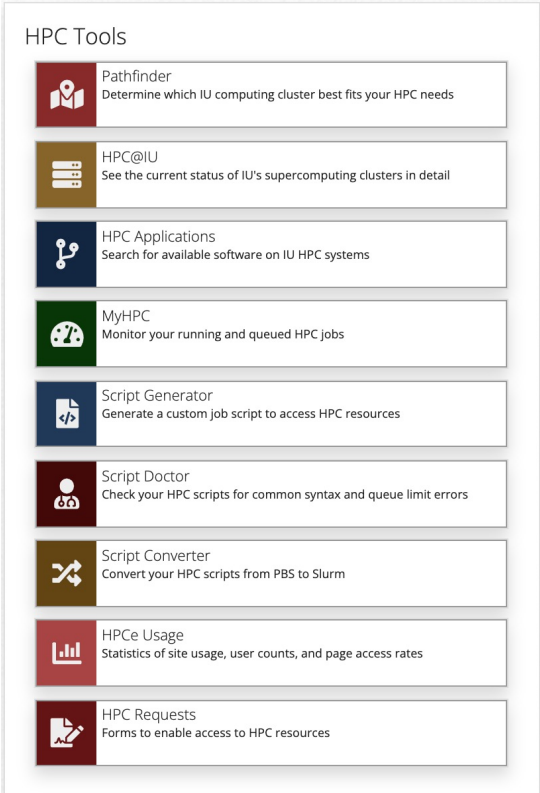
How to write Slurm scripts?

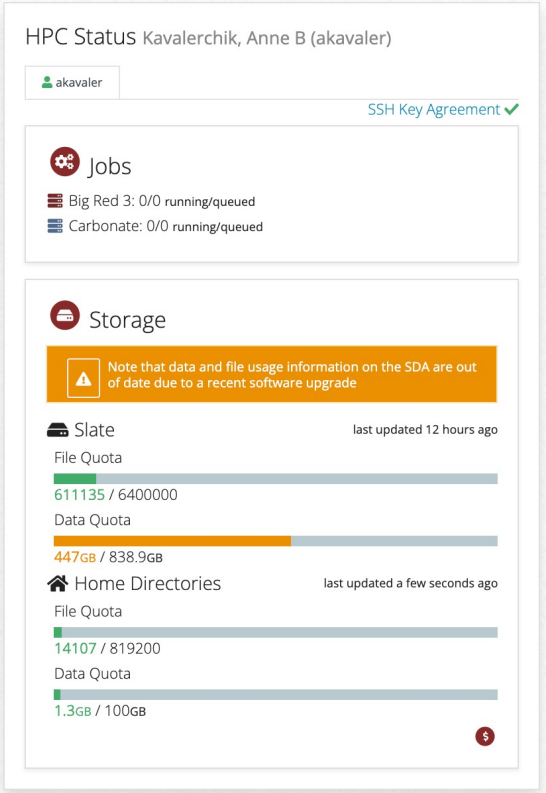
From HPC Everywhere:

Luckily, IU makes this very easy for us!

On HPC Everywhere there is:

- Script Generator
- Script Doctor





Batch job

Which HPC cluster you are using

Whether it is a batch (general) or interactive job

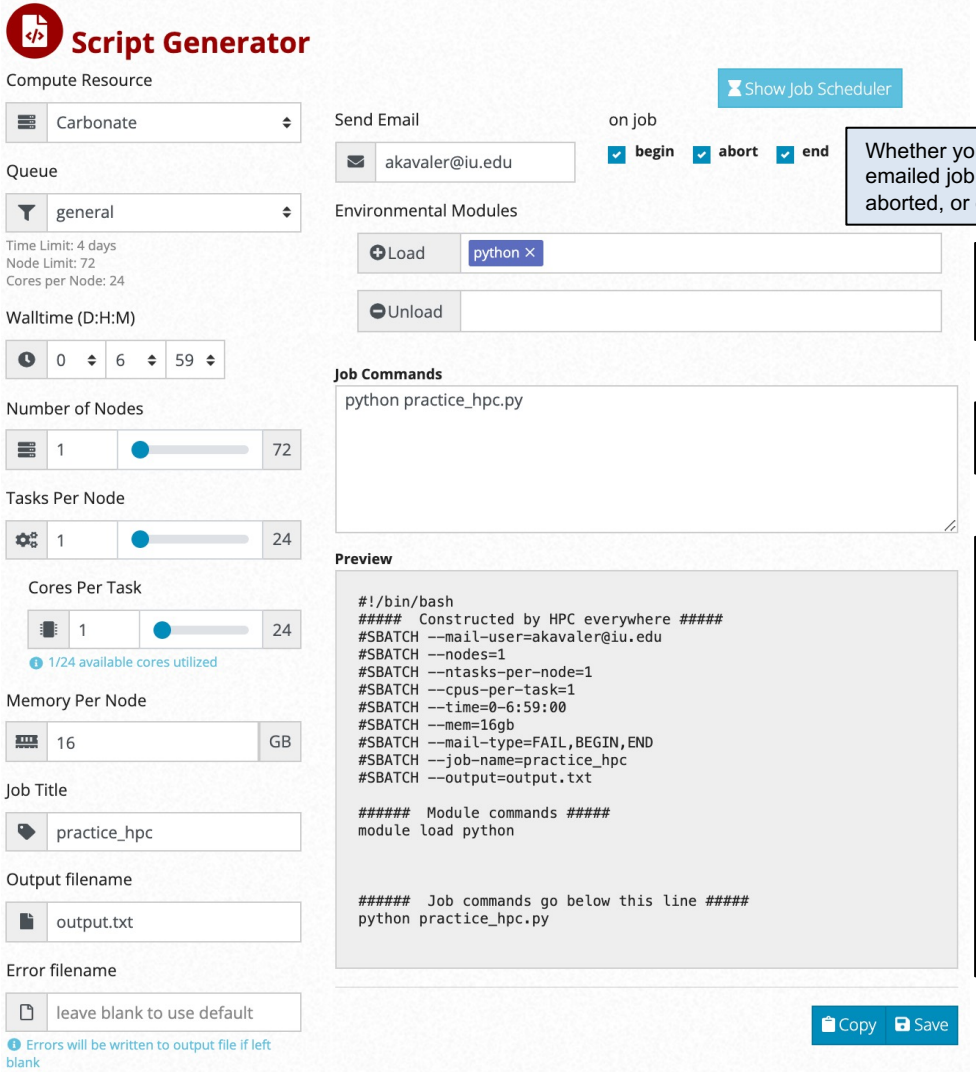
How long you want your job to run

How many nodes, tasks per node, and cores per task – this has to do with parallel computing

How much memory per node to allocate

The name of the job

Filename for output



Whether you want to be emailed job begins, is aborted, or ends

The modules (software) you need for the job

What you want the job to do

The resulting SLURM script! This should be put into a text file and saved with the extension .script, and then it is ready to be executed with `sbatch` and then entered into the job scheduler.

You can log out of the HPC server and your job will run.

Summary

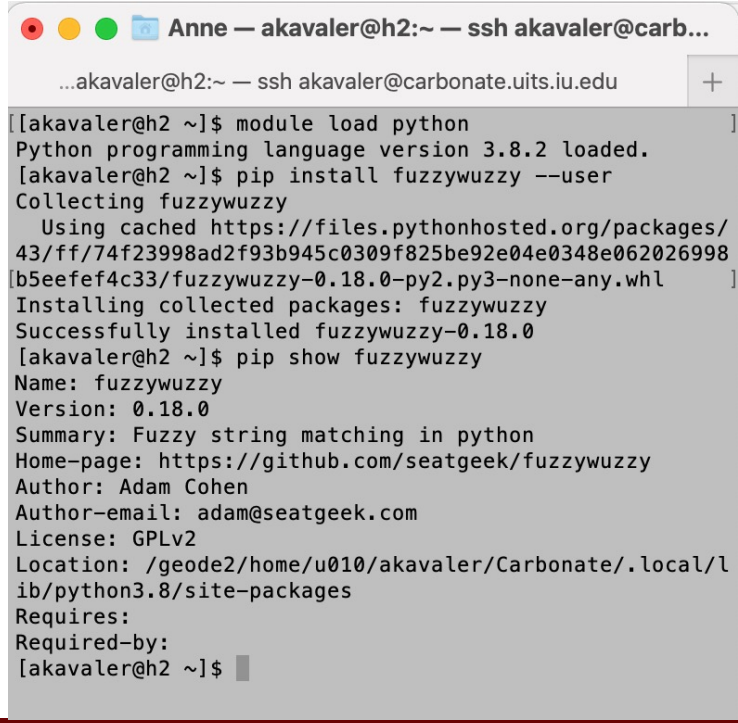
1. `practice_hpc.py` contains the python script that we want to run
2. `practice_slurm.script` is the SLURM script that tells HPC to run the Python script
3. `sbatch practice_slurm.script` will send the Slurm script to the job scheduler which will eventually execute myscript.py
4. `squeue` will show us the queue, `squeue -u [username]` will tell us the status of our job in the queue

Installing additional software: Python

Both the Python and R modules on HPC come pre-installed with many libraries, but may be missing some you need.

You can install these libraries on your home directory on the cluster, or request that it be installed system-wide.

To install a python package in my home directory, I load python and use pip



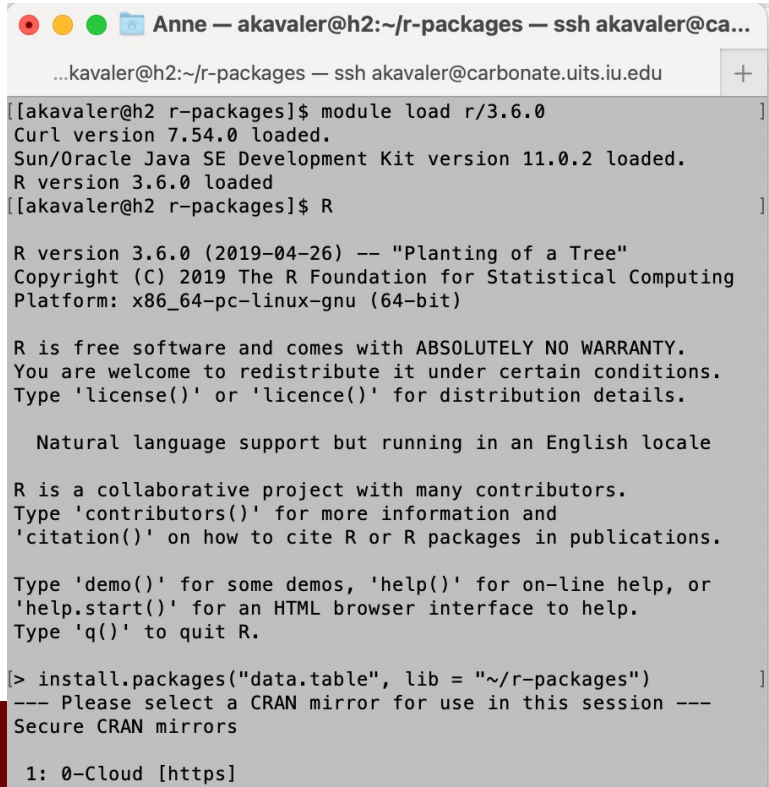
Installing additional software: R

Similarly, we can install R packages in our home directory.

I do the following (adapted from this now archived KB article):

1. Create a directory for your R packages on the home directory (`mkdir ~/r-packages`)
2. Run R in terminal, and install package from CRAN
3. Specify location of package in R script with

```
library("my_r_pkg", lib.loc="~/r-packages/")
```



Part 6

Final remarks

Best practices

Etiquette: **Be a good HPC citizen!**

- Only reserve the HPC resources that you need
- Better to submit many small jobs rather than one large job
- Don’t run jobs on the head node

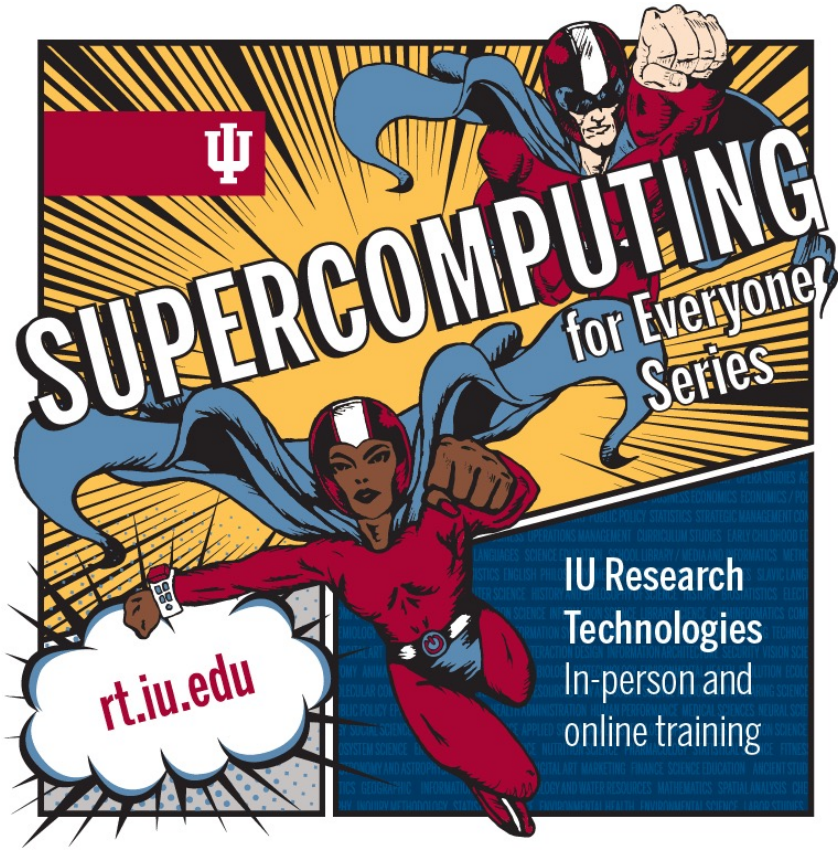
Workflow

- Give jobs, files, and folders descriptive and systematic names
- Make code readable (especially important when working on code in Shell environment)

Additional resources

Supercomputing for Everyone Series (S4ES); a series of on-demand online courses on Canvas and in-person workshops by UITS

- Canvas
  - Introduction to Carbonate
  - HPC Onboarding for Biologists (useful for social scientists too!)
- Upcoming workshop
  - Thursday, 12/2/2021: HPC Workshop run by Research Technologies



Additional resources, continued

LOTS of information on the IU Knowledge Base

- Unix Basics
- Install and load R packages on Big Red II at IU (though now archived, still useful for using Carbonate and other maintained clusters)
- Use Python on IU research supercomputers
- Use SLURM to submit and manage jobs on high performance computing systems

Blogpost on IU HPC: "How to use python on HPC for Indiana University students"

Workshop on using snakemake (for parallel processing in Python) by Kaicheng Yang: Video and Github page

Additional resources, continued

Non-IU sources

- Learn Command Line the Hard Way (it's not that hard!)
- Website: "Introduction to Using the Shell in a High-Performance Computing Context"
- Iowa State University's HPC cluster guide
- University of Maryland Division of Information Technology job submission guide
- Princeton Research Computing's guide to using SSH to connect to clusters
- Brigham Young University Office of Research Computing's guide to submitting multiple jobs at once
- Tennessee Tech's Submitting Groups of HPC Jobs with Job Arrays

Thank you!

My email: akavaler@iu.edu

Research Technologies Support: https://rt.iu.edu/support/index.html

INDIANA UNIVERSITY BLOOMINGTON