







Concepts in Cluster Computing – Introduction to Biocomputing

Ken Weiss

HITS Computational Research Consulting Division

April, 2021

Roadmap

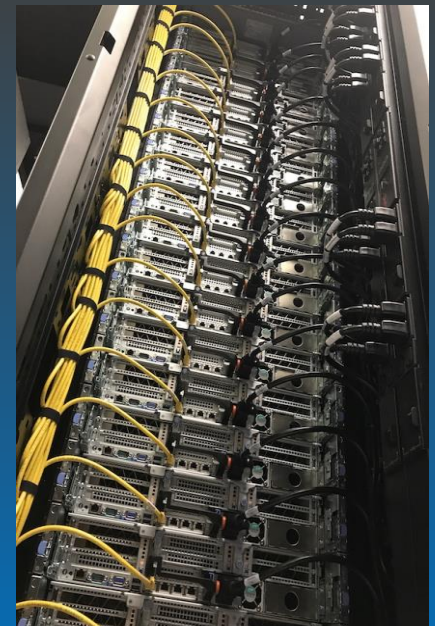
-  Great Lakes overview
-  Software modules
-  Slurm script creation
-  Job submission/monitoring
-  File transfers
-  Advanced topics

Great Lakes

Great Lakes

- Great Lakes is a university-wide shared computational discovery / high-performance computing service used for research.
- Interdisciplinary
 - Provided and operated by Advanced Research Computing at U-M (ARC)
 - Hardware procurement, software licensing, billing support by U-M ITS
 - Used across campus
- Collaborative effort since 2010 between:
 - Advanced Research Computing at U-M (ARC)
 - College of Engineering's IT Group (CAEN)
 - Information and Technology Services (ITS)
 - Medical School (HITS)
 - College of Literature, Science, and the Arts
 - School of Information

<https://arc.umich.edu/greatlakes/>



Using Software on Great Lakes

Access to Software

- Software modules have been installed on Great Lakes to make it easy to use the software which you need. Use the module command to setup the software for your session.

<code>module avail</code>	(list software available for use)
<code>module list</code>	(list the modules currently loaded)
<code>module load <i>software_package</i></code>	(load a software module)
<code>module load <i>collection</i></code>	(load a software collection)
<code>module unload <i>software_package</i></code>	(unload a software module)
<code>module whatis <i>software_package</i></code>	(get a description of the software)
<code>module help <i>software_package</i></code>	(get info about the software)
<code>module spider <i>some_string</i></code>	(search for a module)
<code>module purge</code>	(unload all modules)

- You can create your own modules in your /home directory (if you wish to install software yourself.)
(see <http://arc.umich.edu/software/creating-modules/>)

Slurm Script Creation

Slurm Script Creation

- 🌐 Slurm = Simple Linux Utility for Resource Management
- 🌐 With so many users on the system and a finite number of resources, the scheduler takes job requests and allocates resources, assigns which compute node(s) the job will run on and maintains 'fair sharing' of the cluster.
- 🌐 The scheduler also maintains job logs which are useful for:
 - 🌐 selecting the proper amount of resources to request
 - 🌐 troubleshooting issues with a user's workflow
 - 🌐 identifying hardware issues with the compute nodes

A Sample Slurm Script

Scheduler Directives

```
#!/bin/bash

#SBATCH --job-name=coinflip          ## Name of the job for the scheduler
#SBATCH --account=micro612w21_class  ## generally your PI's username will go here
#SBATCH --partition=standard         ## name of the queue to submit the job to.
                                     ## (Choose from: standard, debug, largemem, gpu)
##SBATCH --gres=gpu:1               ## if partition=gpu, number of GPUS needed
                                     ## make the directive = #SBATCH, not ##SBATCH
#SBATCH --nodes=1                   ## number of nodes you are requesting
#SBATCH --ntasks=1                  ## how many tasks (resource spaces) do you want to reserve
#SBATCH --cpus-per-task=1           ## how many cores do you want per task
#SBATCH --time=00:30:00             ## Maximum length of time you are reserving the resources for
                                     ## (if job ends sooner, bill is based on time used)
#SBATCH --mem=2g                    ## Memory requested for this job
#SBATCH --mail-user=kgweiss@umich.edu ## send email notifications to umich email listed
#SBATCH --mail-type=END              ## when to send email (standard values are:
                                     ## NONE, BEGIN, END, FAIL, REQUEUE, ALL.
                                     ## (See documentation for others)
#SBATCH --output=.%x-.%j            ## send output and error info to the file listed
                                     ##(optional: different name format than default)
```

```
# With SLURM, you can load your modules in the SBATCH script
module load R
```

```
# optional but very useful for support staff
my_job_header
```

```
# Put your job commands after this line
R CMD BATCH --no-save --no-restore coinflip.R
echo -en "number of heads is:"; grep H headtail.txt | wc -l
echo -en "number of tails is:"; grep T headtail.txt | wc -l
```

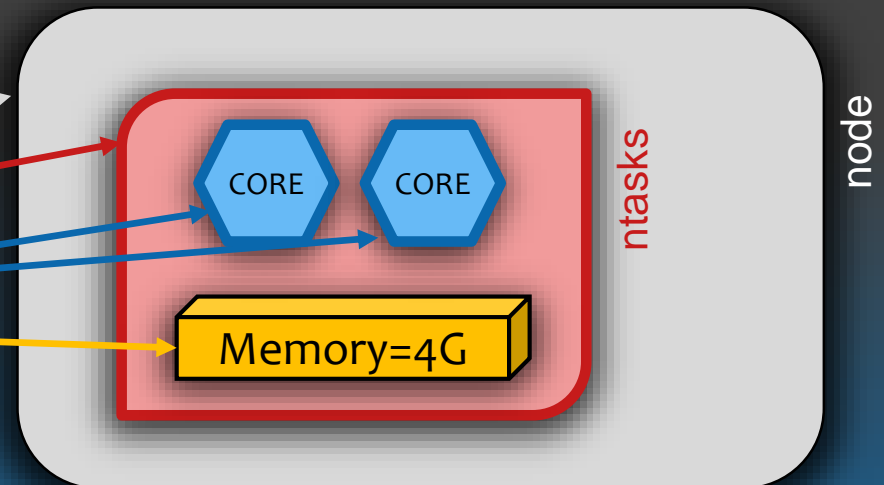
← **Your workflow goes here**

For a detailed description of scheduler directives go to:

<https://arc.umich.edu/greatlakes/slurm-user-guide/>

Basic Resource Request (single node example)

```
#!/bin/bash
##### Slurm preamble
#SBATCH --job-name=fmm_test
#SBATCH --mail-type=END,FAIL
#SBATCH --account=test
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=2
#SBATCH --mem=4G
#SBATCH --partition=standard
#SBATCH --time=1:00:00
#SBATCH --output=./%x-%j
##### End of preamble!
```



/scratch

- 🌐 What is /scratch?
/scratch is a high performance file system connected to fast networking which allows for high throughput reads/writes
- 🌐 Why should I use /scratch?
In a word... performance
- 🌐 What should I do with my files on /scratch
Once your program is done running, copy your results files to your /home directory or other storage you have access to
- 🌐 Your /scratch space for this class is:
`/scratch/micro612w21_class_root/micro612w21_class/username`
- 🌐 WARNING: /scratch is NOT backed up. Make sure that files contained on /scratch are not your only copy.

Job Submission

Job Submission/Monitoring

- For a job to run on the Great Lakes cluster, you must submit the job to the scheduler.
- On Great Lakes, the command to send jobs to the scheduler is:

`sbatch Slurm_script_name`

- Useful commands for working with queued jobs:

`scancel <job ID #>` - removes the job from the queue

(will work also if the job is running)

`squeue -u <uname>` - lists your jobs and their status in the queue

`sq` – system defined shortcut for `squeue -u <uname>`

`scontrol show job <job ID #>` - get info on your job (mileage varies on running jobs)

`my_job_statistics <job ID #>` - get stats on a completed job

Job Submission Example

- 🌐 First, let's get the files into your /scratch directory.
type: `cd /scratch/micro612w21_class_root/micro612w21_class/username`
`cp ../kgweiss/coinflip.* .`
- 🌐 Next, modify coinflip.sbat to send email to your username.
type: `nano coinflip.sbat`
modify: `#SBATCH --mail-user username@umich.edu`
save and exit
- 🌐 Finally, submit the job
type: `sbatch coinflip.sbat`

File Transfers

File Transfers







- 🌐 Eventually, you will need to copy data either to or from a remote server/cluster (in this particular case: Great Lakes)
- 🌐 To do so, use a secure copy application (demo WinSCP)
- 🌐 PC users can use:
 - WinSCP
 - Cyberduck
 - Filezilla
- 🌐 Mac users can use:
 - Fetch
 - Cyberduck
 - Filezilla
- 🌐 Please visit: <https://its.umich.edu/computing/computers-software/compute> to obtain secure copy software
- 🌐 On Great Lakes, use the hostname: greatlakes-xfer.arc-ts.umich.edu for file transfers

Advanced Topics

Advanced Topics

- 🌐 Job dependencies – Start a job after some condition
- 🌐 Job arrays – Submit one job for multiple data files
- 🌐 Interactive jobs – Use a command line on a compute node
- 🌐 wget - Grab files from websites
- 🌐 Open OnDemand – Using Great Lakes with a graphical interface
See <https://greatlakes.arc-ts.umich.edu> for details
- 🌐 Globus - another, very useful way to copy files (web based)
See <http://globus.org/> for details

Useful Links

-  <https://arc.umich.edu/>
-  <https://arc.umich.edu/greatlakes/slurm-user-guide/>
-  <http://arc.umich.edu/software/>
-  https://twitter.com/ARCTS_UM/
-  <http://its.umich.edu/accounts-access/uniqnames-passwords/two-factor-authentication>
-  <https://greatlakes.arc-ts.umich.edu/>
-  <https://globus.org/>

Contact Info

-  arcts-support@umich.edu
-  Personal Contact Information
 -  kgweiss@umich.edu