

## Parallel Computing in SLURM with Mathematica

### Interactive Jobs:

Interactive jobs are good for testing out scripts before you send them off in batch or running small bits of code. Yale's cluster "Grace" allows users to utilize up to 4 cores on the interactive partition. To do interactively run a job in Mathematica:

1. Write your script in Mathematica on your own machine. You can write code while on the cluster's interactive interface but it is prohibitively slow. Be sure to include "LaunchKernels[4]" at the beginning of your script. This will launch 4 parallel cores to run your code on.
2. Open a new terminal window and copy your notebook to your cluster account.

```
scp /Users/ericaholdridge/Desktop/projects/testmath.nb  
user@grace.hpc.yale.edu:~/
```

3. Login to the cluster using the -Y modifier, which lets the cluster know you would like to use X11.

```
ssh -Y user@grace.hpc.yale.edu
```

Note: If you have not already installed the X-Quartz graphical interface, you will need to download the version here for Grace: <https://www.xquartz.org/releases/XQuartz-2.7.8.html>. To make sure it is working once you are on Grace, you can run the command "xclock", which should open a new X11 window with a clock.

4. To begin your interactive session run:

```
srun --pty --x11 -c 4 -p interactive -t 2:00:00 bash
```

Specify the number of cores you want to use after -c and the amount of time you will need after -t. The maximum cores on Grace's interactive partition is 4 and the maximum wall time is 6 hours (see <https://docs.ycrc.yale.edu/clusters-at-yale/clusters/grace/> for more info on Grace's partitions).

5. Load and launch Mathematica:

```
module load Mathematica/11.3.0  
mathematica
```

If you want to change the available versions of Mathematica enter:

module avail mathematica

6. An interactive window that looks more or less like Mathematica on your own machine will (slowly) appear. Click “Open”, navigate to where your .nb file is under your user name, click on it and click “Open”. Your notebook will (slowly) appear.
7. At the top of the window click “Evaluation”>“Evaluate Notebook”. Your notebook should run. If you receive an error that there are no parallel cores available be sure that you (a) included “LaunchKernels[4]” at the top of your script and (b) have parallel local kernels configured by going to “Evaluation”>“Parallel Kernel Configuration” and clicking on the “Local Kernels” tab. If it is not, click “Enable Local Kernels”.

### Submitting Batch Jobs

Once you know your code plays nicely with Grace, you can submit it as a batch job. This will allow you to take advantage of more cores and more computing power.

1. The shell script is what will tell Grace how you want to allocate resources and submit your Mathematica script. The specifics of what you will include here is going to vary depending on what you’re trying to do, but here is the gist:

```
#!/bin/bash

#SBATCH --job-name=project
#SBATCH --ntasks=20
#SBATCH --ntasks-per-node=20
#SBATCH --partition=day
#SBATCH --mail-type=ALL
#SBATCH --time=24:00:00

module load Mathematica/11.3.0
math -run < testmodel.txt
```

The first line is necessary to tell the cluster that it’s reading a bash script. The next several lines are what will change substantially. Line 2 gives the job a name. Line 3 tells it that I want to use 20 cores. Different nodes on Grace have different numbers of cores. You can familiarize yourself with the hardware here <https://docs.ycrc.yale.edu/clusters-at-yale/clusters/grace/> . Line 4 tells it how many cores per node I want to use. In this case, I wanted to keep all of my cores on the same node because the particular script I am running requires that cores “talk to each other” (this doesn’t work so well if they’re split across nodes). Line 5 specified the partition to be used. Grace’s “day” partition has a maximum wall time of 24 hours. The link above gives details about that and other partitions. Line 6 says that I want to receive an email if the job ends, fails, etc. Line 7 specifies the wall time that I want to use. If the job finishes before time is up, it will exit no problem. If it’s not done, Grace will kill the job where it is. For this reason, it’s a good idea to write code that saves output through the evaluation.

The last couple lines load Mathematica and then launch or Mathematica script.

2. Once you're satisfied with your script, you need to save it. Go to "Format">"Make Plain Text". Then save your file under whatever name you like.
3. Be sure to uncheck the box that says "if no extensions provided use '.txt'".
4. Navigate to where this file is in Finder, right click and select "Get Info". Click on the lock on the bottom right corner, enter the password you use to login to your computer. This will unlock the file.
5. Copy your shell script and your Mathematica script to your Grace account. You can copy each of them separately or copy an entire folder containing both files using:

```
scp -r /Users/user/Desktop/modelfolder user@grace.hpc.yale.edu:~/
```

Note: If you copy them as a folder, remember to indicate the correct directory to your Mathematica script in your shell script and when submitting the job. For example:

```
math -run < ~/modelfolder/testmodel.txt
```

6. Login to Grace as you normally would (ssh user@grace.hpc.yale.edu then enter your password). Now you will need to make these files executable. To do so:

```
chmod u+x -R ~/modelfolder
```

The "-R" is only necessary if you're working with a folder rather than a singular file.

7. Submit your job simply by entering:

```
sbatch ~/modelfolder/shellscript.sbatch.txt
```

where shellscript.txt is the shell script you wrote earlier and copied to Grace.

8. If you want to check on the status of your jobs you can enter:

```
squeue -u$USER
```