Running Jobs (Slurm)

Queueing System

Greenplanet uses the Slurm queue and job scheduler. We are currently running version 20.02.5, which has reasonably easy-to-follow documentation. There is also a Cheat Sheet of command options. A nice introduction is available from the Slurm User Group 2017 conference.

If you are used to different system (PBS, LoadLeveler, SGE, etc.), see this PDF that shows corresponding command line tools and job script options. Wrapper scripts for PBS commands (qstat, qsub, etc.) are also installed, which will let you use job submission scripts from an ancient time when we ran Torque/Maui.

You can use the **`sview`** command to open the Slurm GUIFor a list of partitions, look at this page.

If you need to run an interactive shell as a job, for instance to run matlab or test other code interactively, you should use this command (leave off --x11 if X11 graphics are not required):

```
srun --pty --x11 -t 300 -n 1 -p ilg2.3 bash -i
```

(-t is walltime, adjust as required. -p is partition which is the queue name or a comma seperated list of partitions. If you need to run something on a different partition you can replace **`ilg2.3`** with the desired partition.

For non-interactive jobs the normal way to run a job is with the **`sbatch`** command supplied by the path to a slurm job script.

Partitions

Some partitions are limited to certain groups (see Notes column)

Greenplanet Partitions (gplogin2/gplogin3)
[Partitions in blue are best for general use]
[Partitions in red are only on the side being upgraded to Rocky 8/Alma 9 (gplogin1)]

Partition	Nodes	CPU Arch	GFLOPS/core	Cores/node	RAM	Disk	Notes
atlas	**8**	**Intel (Westmere) 2.4/2.9 GHz**	**~11**	**8-12**	**24G**	**0.2-7T**	**ATLAS (Taffard, Whiteson)**
brd2.4	**6**	**Intel (Broadwell) 2.4 GHz**		**28 (4 nodes) 20 (2 nodes)**	**128G (28C) 256G (20C)**	**372G (28C) 1.1T (20C)**	**c-14-10,c-14-12 (28C) c-19-[411-414] (20C)**
cas2.5	4	Intel (Cascade Lake) 2.5 GHz		40	192G	400G SSD	Wodarz
cas3.6	1	Intel (Cascade Lake) 3.6 GHz		16	1.5T	1.7T SSD	Primeau
has2.5	**4**	**Intel (Haswell) 2.5 GHz**		**24**	**128G**	**372G**	
ilg2.3	**46**	**AMD (Interlagos) 6276 2.3 GHz**	**18.4**	**32***	**128G**	**900G**	***32 FP cores, 64 Integer cores**
m-c1.9	10	AMD (Magny-Cours) 6168 1.9 GHz	7.6	48	64G	900G	
m-c2.2	2	AMD (Magny-Cours) 6174 2.2 GHz	8.8	48	64-128G	1.4T	

Partition	Nodes	CPU Arch	GFLOPS/core	Cores/node	RAM	Disk	Notes
nes2.8	**214**	**Intel (Nehalem/ Westmere) 2.7/2.8 GHz**	**~11**	**8-12**	**12-48G**	**100-500G**	
sib2.9	**45**	**Intel (Sandy-/Ivy-Bridge) 2.8/2.9 GHz**	**~23**	**16-20**	**32-128G**	**900G+**	
sky2.4	13	Intel (Skylake) 2.4 GHz		40	96G	220-740G	Moore & Randerson nodes
sky3.0	7	Intel (Skylake) 3.0 GHz		24	384G	1.6T SSD	Lowengrub
wodarz	4	Intel (Broadwell) 2.4 GHz		28	128G		Wodarz
gpu	**2**	**Intel (Haswell) 2.4 GHz + 8 Nvidia TitanX GPUs**					**Mobley, Poulos**
knl1.3	**2**	**Intel (Knights Landing)**		**64**	**48G**		**4 threads/core, no Infiniband**

There is also a special partition called "scavenge" that is intended to make use of otherwise idle nodes. Scavenge contains all the free-access nodes, but jobs may get preempted (killed) by high priority jobs. It must be used with the qos "scavenger".

Batch Matlab

This is example is based on using gplogin2, the login node for the new side of cluster.

We use lmod on gplogin2/3 as opposed to environment-modules on gplogin1.

All examples below are created in your home directory via copy-and-paste into your GP shell account:

Consider the following example, matlab_example.m

```
cat << 'EOF' > ~/matlab_example.m
[X,Y] = meshgrid(-2:.2:2);
Z = X .* exp(-X.^2 - Y.^2);
surf(X,Y,Z);
print('example-plot','-dpng');
exit;
EOF
```

> Running the above matlab example ***WITHOUT*** Slurm: (this is how many people run on the login node which is **BAD**!)

```
cat << 'EOF' > ~/run.sh
#!/bin/bash
```

ml purge ml matlab/R2017b

matlab -nodisplay -nodesktop -nosplash < matlab_example.m EOF

chmod 755 ~/run.sh && ~/run.sh

> running the above matlab example ***WITH*** Slurm:

```
cat << 'EOF' > ~/runv2.sh
#!/bin/bash
```

#SBATCH --job-name=my_matlab_job #SBATCH --output=my_matlab_job.out #SBATCH --error=my_matlab_job.err #SBATCH --partition=brd2.4,has2.5,ilg2.3,m-c1.9,m-c2.2,nes2.8,sib2.9 #SBATCH --time=00:01:00 #SBATCH --nodes=1 #SBATCH --ntasks=16

ml purge ml matlab/R2017b

matlab -nodisplay -nodesktop -nosplash < matlab_example.m EOF

chmod 755 ~/runv2.sh

To submit the job

```
sbatch runv2.sh
```

This will create a file in your home directory

```
example-plot.png
```

The hardest part is determining how much resources your computation/simulation will need.

One has to pick an partition based on the computation. Usually people will want Intel CPUs, but we have AMD CPUs as an option.

- Partitions

The resources for your computation/simulation needs to be determined emperically.

Method 1: Make a Slurm submission script and guestimate your resources required (CPU cores, number of nodes, walltime etc.).

Submit the job via sbatch and then analyze the effiency of the job with seff and refine your scheduling parameters on the next run.

```
$ seff 10773
Job ID: 10773
Cluster: blueplanet
User/Group: santucci/staff
State: COMPLETED (exit code 0)
Nodes: 1
Cores per node: 32
CPU Utilized: 00:00:20
CPU Efficiency: 1.56% of 00:21:20 core-walltime
Job Wall-clock time: 00:00:40
Memory Utilized: 448.57 MB
Memory Efficiency: 2.74% of 16.00 GB
```

If you want to see which node was selected for the job look at the epilog output

```
$ cat slurm.epilog-10773
-------------- slurm.epilog ---------------
```

```
Job ID:    10773
User:      santucci
Group:     staff
Job Name:  my_matlab_job
Partition: has2.5
QOS:       normal
Account:   staff
Reason:    None,c-19-293
Nodelist:  c-19-293
Command:   /data11/home/santucci/runv2.sh
WorkDir:   /data11/home/santucci
BatchHost: c-19-293
```

Method 2: Request an interactive shell and experiment to determine how much memory is required and how long it needs to run.

```
srun --pty --x11 -t 300 -n 1 -p <partition-list> bash -i
```

recommendations on profile are available @
https://www.nccs.nasa.gov/user_info/slurm/determine_memory_usage

If new to Slurm please see this page.

Here are two quick reference guides that you will want to have handy:

https://slurm.schedmd.com/pdfs/summary.pdf

https://www.chpc.utah.edu/presentations/SlurmCheatsheet.pdf

Credit: inspiration for this example comes from https://it.math.ncsu.edu/hpc/slurm/batch/matlab

Interactive Matlab

Remember gplogin* (login node) is a shared resource, so one shouldn't be running matlab directly on a login node. It's not a problem until someone points it out as a problem, so we need everyone to get in the habit of requesting an interactive session for Matlab when not using the Slurm scheduler.

for interactive sessions on Intel

```
srun --pty --x11 -t 24:00:00 -n 1 --mem=8192 -p brd2.4,has2.5,nes2.8,sib2.9,sky2.4 bash -i
```

for interactive sessions on AMD

```
srun --pty --x11 -t 24:00:00 -n 1 --mem=8192 -p ilg2.3,m-c1.9,mc2.2 bash -i
```

after you get a shell

```
ml purge
ml matlab
matlab &
```