SLURM
Job Scrtipts on

HiPerGator

The University of Florida Supercomputer for Research

Matt Gitzendanner magitz@ufl.eduv

UF | Information Technology

OneIT for the #GatorGood

HiPerGator 2.0

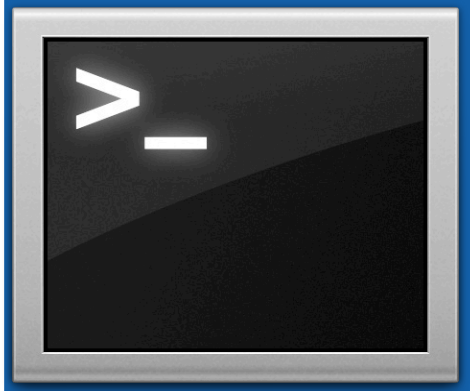
The University of Florida Supercomputer

UF | Information Technology

OneIT for the #GatorGood

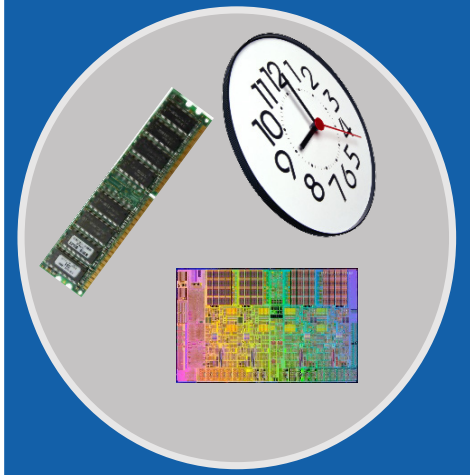
Cluster Basics

User interaction



Login node (Head node)

User interaction



User interaction

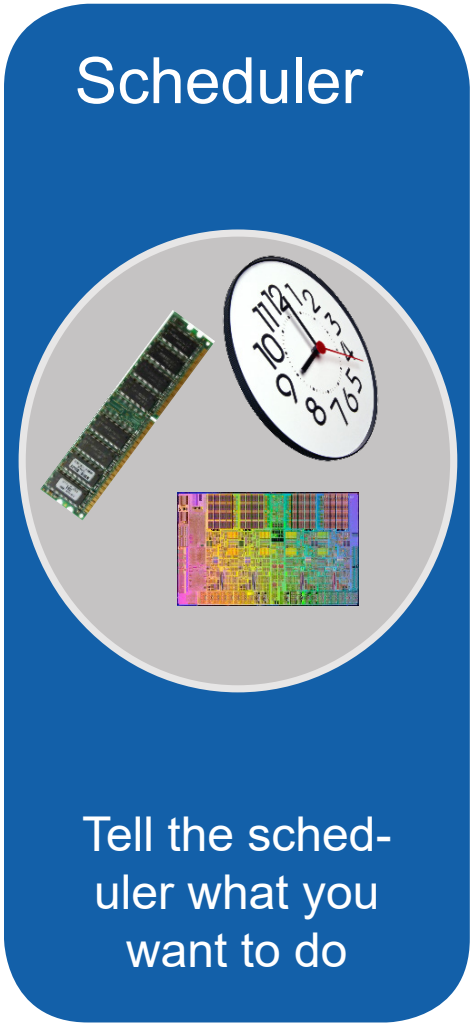
User interaction

User interaction

Scheduling a job

- Need to tell scheduler what you want to do
  - How many CPUs you want and how you want them grouped
  - How much RAM your job will use
  - How long your job will run

  The commands that will be run



Basic SLURM job script

```
#!/bin/sh
#SBATCH --job-name=serial_job_test   # Job name
#SBATCH --mail-type=ALL              # Mail events
#SBATCH --mail-user=email_address    # Where to send mail
#SBATCH --ntasks=1                   # Run on a single CPU
#SBATCH --mem=1gb                    # Memory limit
#SBATCH --time=00:05:00              # Time limit hh:mm:dd
#SBATCH --output=serial_%j.out       # Output and error log

pwd; hostname; date
module load python
echo "Running plot script on a single CPU core"
python /ufrc/data/training/SLURM/plot_template.py
date
```

SLURM CPU Requests

- Nodes: --nodes or -N
  - Request a certain number of physical servers
- Tasks: --ntasks or -n
  - Total number of tasks job will use
- CPUs per task: --cpus-per-task or -c
  - Number of CPUs per task

HiPerGator 2.0 Servers (30,000 cores):

32 cores (2 X 16-core Intel Xeon CPUs)

HiPerGator 1 Servers (16,000 cores):

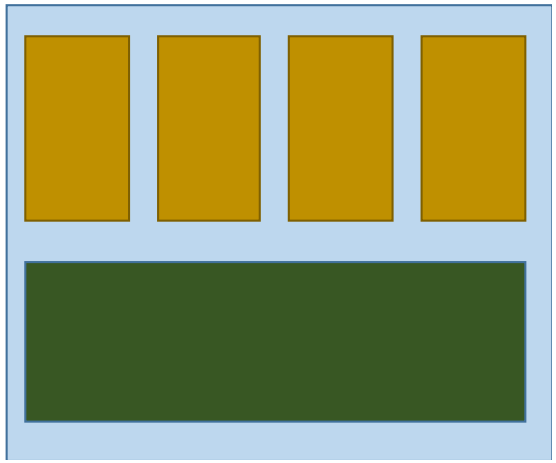
64 cores (4 X 16-core AMD CPUs)

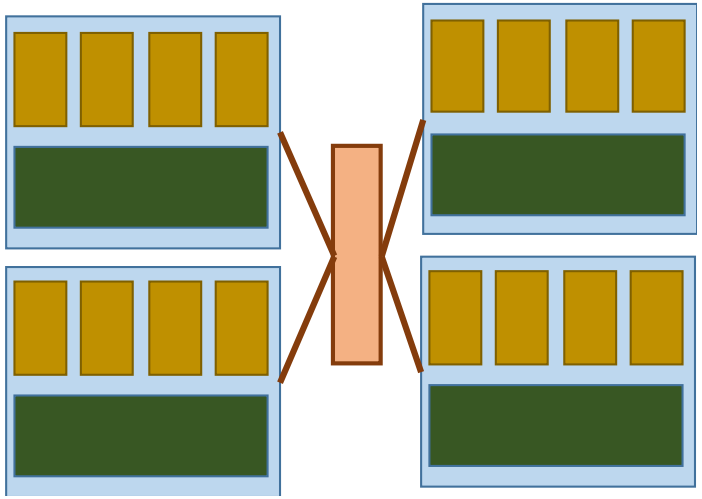
SLURM CPU Requests

- For single processor jobs
  - `#SBATCH --nodes=1`
  - `#SBATCH --ntasks=1`
  - `#SBATCH --cpus-per-task=1`

SLURM CPU Requests

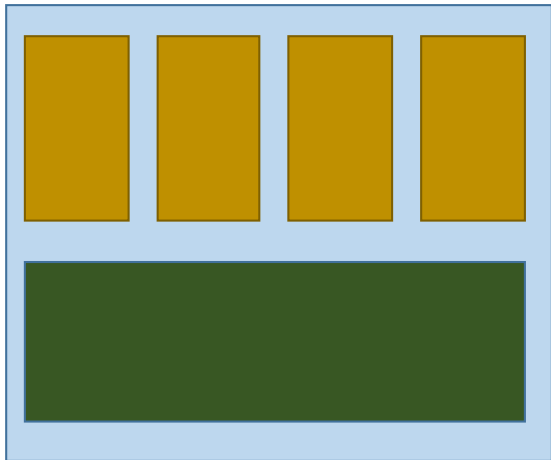
- Parallel applications
  - OpenMP, Threaded, Pthreads
    - All cores on one sever, shared memory
  - MPI
    - Can use multiple servers





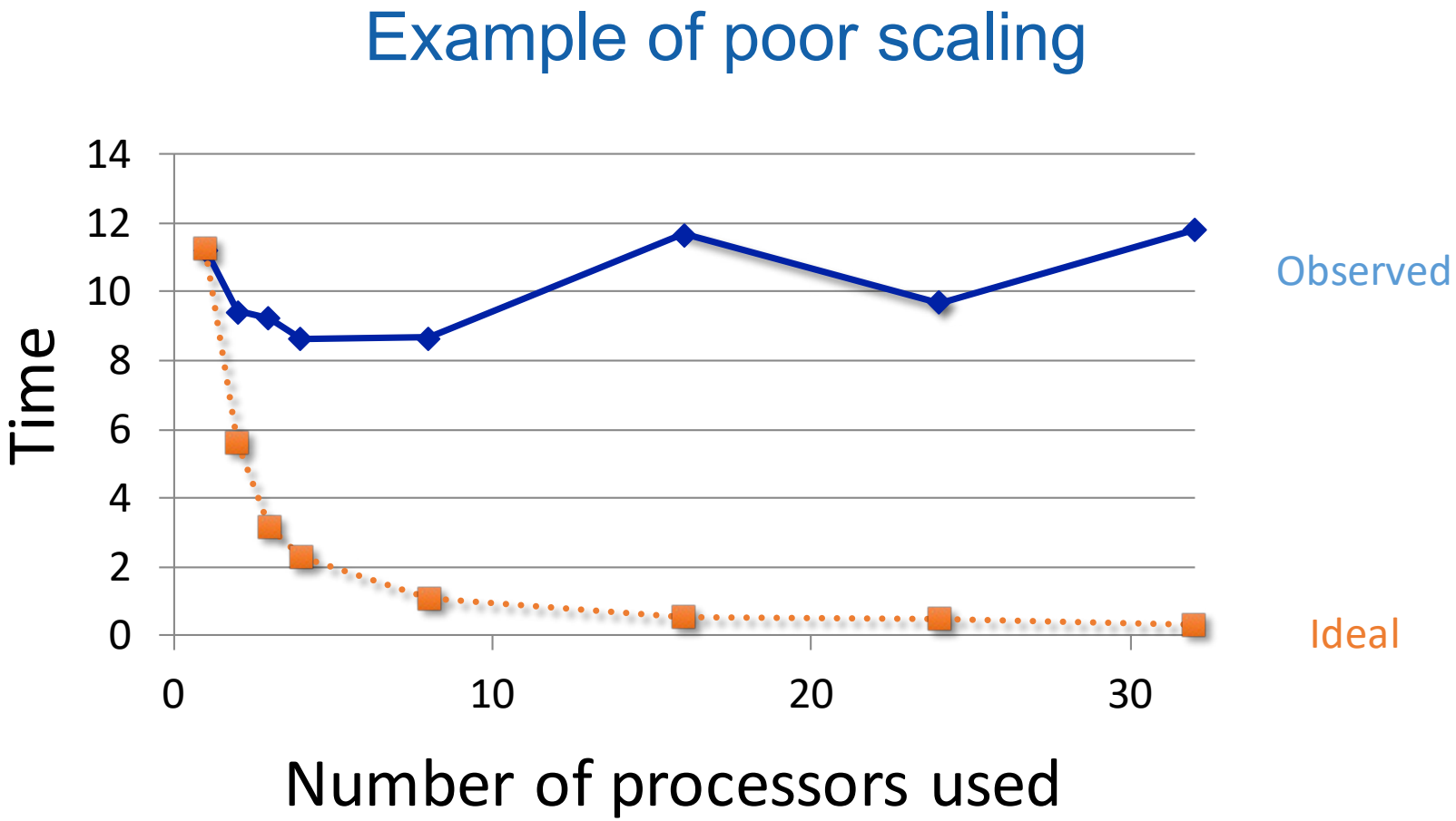
SLURM CPU Requests

- For threaded applications (single node):
  - `#SBATCH --nodes=1`
  - `#SBATCH --ntasks=1`
  - `#SBATCH --cpus-per-task=8`



Parallel efficiency

- How well does your application scale?



SLURM Memory Requests

- Memory:

```
#SBATCH --mem-per-cpu=1gb
```

Or

```
#SBATCH --mem=1gb
```

  - Can use mb or gb
  - No decimal values: use 1500mb, not 1.5gb

HiPerGator 2.0 Servers:
~120 GB RAM
HiPerGator 1 Servers:
256GB RAM

Emails

```
Job ID: 94392
Cluster: hipergator
User/Group: magitz/ufhpc
State: COMPLETED (exit code 0)
Nodes: 1
Cores per node: 4
CPU Utilization: 00:00:44
CPU Efficiency: 52.38% of 00:01:24 core-walltime
Memory Utilization 1.52 MB
Memory Efficiency: 0.04% of 4.00 GB
```

Emails

```
Job ID: 5019
Cluster: hpg1
User/Group: magitz/ufhpc
State: CANCELLED (exit code 0)
Cores: 1
CPU Utilization: 00:00:00
CPU Efficiency: 0.00% of 00:00:00 core-walltime
Memory Utilization 1.26 MB
Memory Efficiency: 126.17% of 1.00 MB
```

Job error file:

```
slurmstepd: Job 5019 exceeded memory limit (1292 > 1024), being killed
slurmstepd: Exceeded job memory limit
slurmstepd: *** JOB 5019 ON dev1 CANCELLED AT 2016-05-16T15:33:27 ***
```

SLURM Time Request

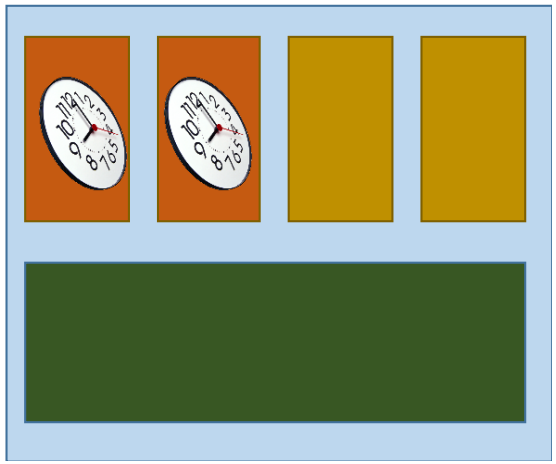
- Time: `--time` or `-t`

```
#SBATCH --time=2:00:00
```

  - 120 (minutes)
  - 2:00:00 (hh:mm:ss)
  - 7-0 (days-hours)
  - 7-00:00 (days-hh:mm)
  - 7-00:00:00 (days-hh:mm:ss)

SLURM Time Request

- Limits:
  - Investment QOS: 31 days
  - Burst QOS: 4 days
  - Dev partition: 12 hours
  - GUI partition: 96 hours



As with all resource requests, providing a reasonably accurate request ensures best results

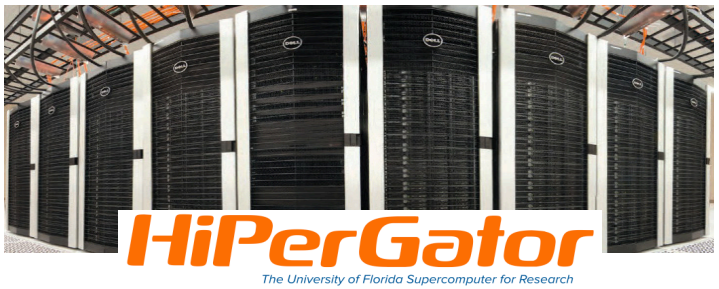
Quality of Service (--qos)

- Each group has two QOS options
  - Investment QOS:
    - The NCUs the group has purchased
    - `--qos=group` (or leave off as this is default)
  - Burst QOS:
    - The burst capacity, available when idle resources are available on the cluster
    - `--qos=group-b`
- Users can choose higher priority, or larger pool of resources

Partition (--partition or -p)

hpg2-compute (30,000 cores)

- 32 Intel cores/server
- Default partition
- 75-90% utilized



hpg1-compute (16,000 cores)

- 64 AMD cores/server
- -p hpg1-compute
- 0-5% utilized

SLURM output/error files

```
#SBATCH -o output.file
#SBATCH -e error.file
#SBATCH -o output.file #W/o -e
```

combined

- Can also use --output and --error

```
#SBATCH --output JobFile.%j.out
```

- Use %j instead of $SLURM_JOBID

SLURM Task Arrays

- `#SBATCH --array=1-200%10`
  - Task range with % to limit number of jobs at a time

- `$SLURM_ARRAY_TASK_ID`
- Output file naming:
  - %A: job id
  - %a: task id
  - Output.%A_%a.out

Multiple groups

- Some users are members of multiple groups

```
#SBATCH --account=group
#SBATCH --qos=group

#SBATCH --account=group
#SBATCH --qos=group-b
```

SLURM

- Note that multi-letter directives are double-dash:
  - `--mail-type`
  - `--ntasks`
  - `--mem-per-cpu`

```
sbatch: error: distribution type
'ail-type=ALL' is not recognized
```

- Do not use spaces with =
  - `--mail-user=magitz@ufl.edu` ✔
  - `--mail-user magitz@ufl.edu` ✔
  - not: `--mail-user= magitz@ufl.edu`

SLURM environment

- SLURM inherits your environment
  - This includes present working directory
    - Don’t need cd $SLURM_SUBMIT_DIR
  - Modules that are loaded
  - Be careful of conflicting modules

Submitting and checking on jobs

- `sbatch job_file.sbatch`
- `squeue -u username`
- `sacct`
- See wiki.rc.ufl.edu/doc/SLURM_Commands
- See http://slurm.schedmd.com/

Development sessions

- `module load ufrc`
- Followed by
  - `srundev`
  - `srundev -t 60`
  - `srundev -t 60 -c 4`

Example files

```
cd /ufrc/group/user/
mkdir SLURM_examples
cd SLURM_examples
cp /ufrc/data/training/SLURM/*.sbatch .
```

Support

Support requests

Web page and wiki

HiPerGator 2.0 Information

- HiPerGator 2.0 Information
- SLURM Documentation
- Moab (PBS) to SLURM command reference