The Agave Cluster

Gil Speyer speyer@asu.edu

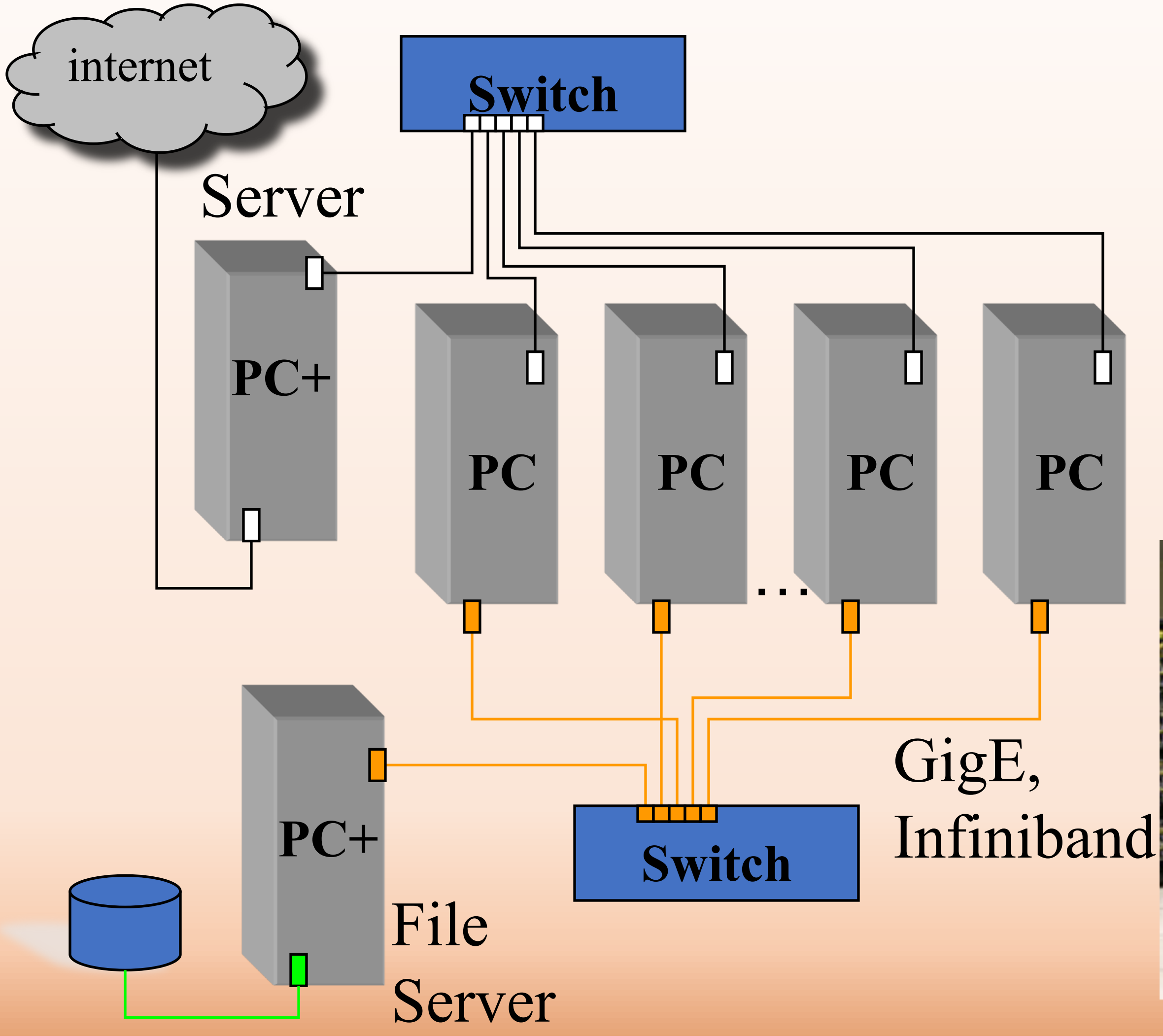
February 11, 2019

Knowledge Enterprise Development
ARIZONA STATE UNIVERSITY

Outline

- System information
- Initial login
- Transferring files
- Modules
- Batch System
- Job Monitoring
- Interactive mode
- Other resources
- Good citizenship

Generic Cluster Architecture

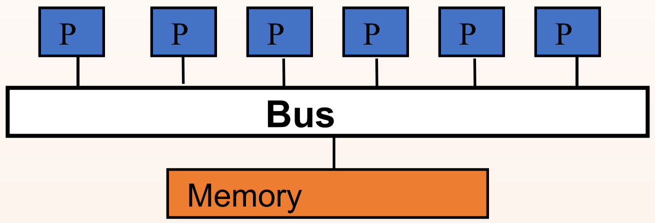


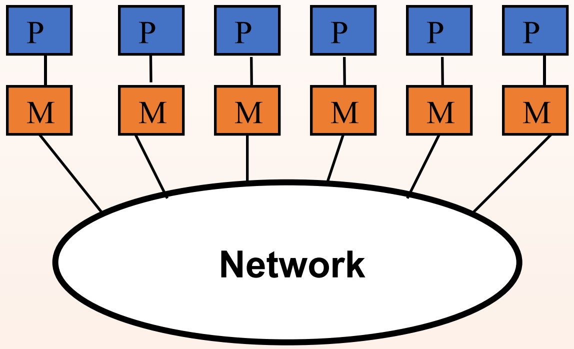
(Adv. HPC System)

lonestar.tacc.utexas.edu

ASU Knowledge Enterprise Development
ARIZONA STATE UNIVERSITY

Shared and Distributed Memory





Shared memory (SMP): single address space. All processors have access to a pool of **shared memory**. Agave normal and phi nodes have 28 and 64 cores per node, respectively.

Methods of memory access :
- Bus, Crossbar

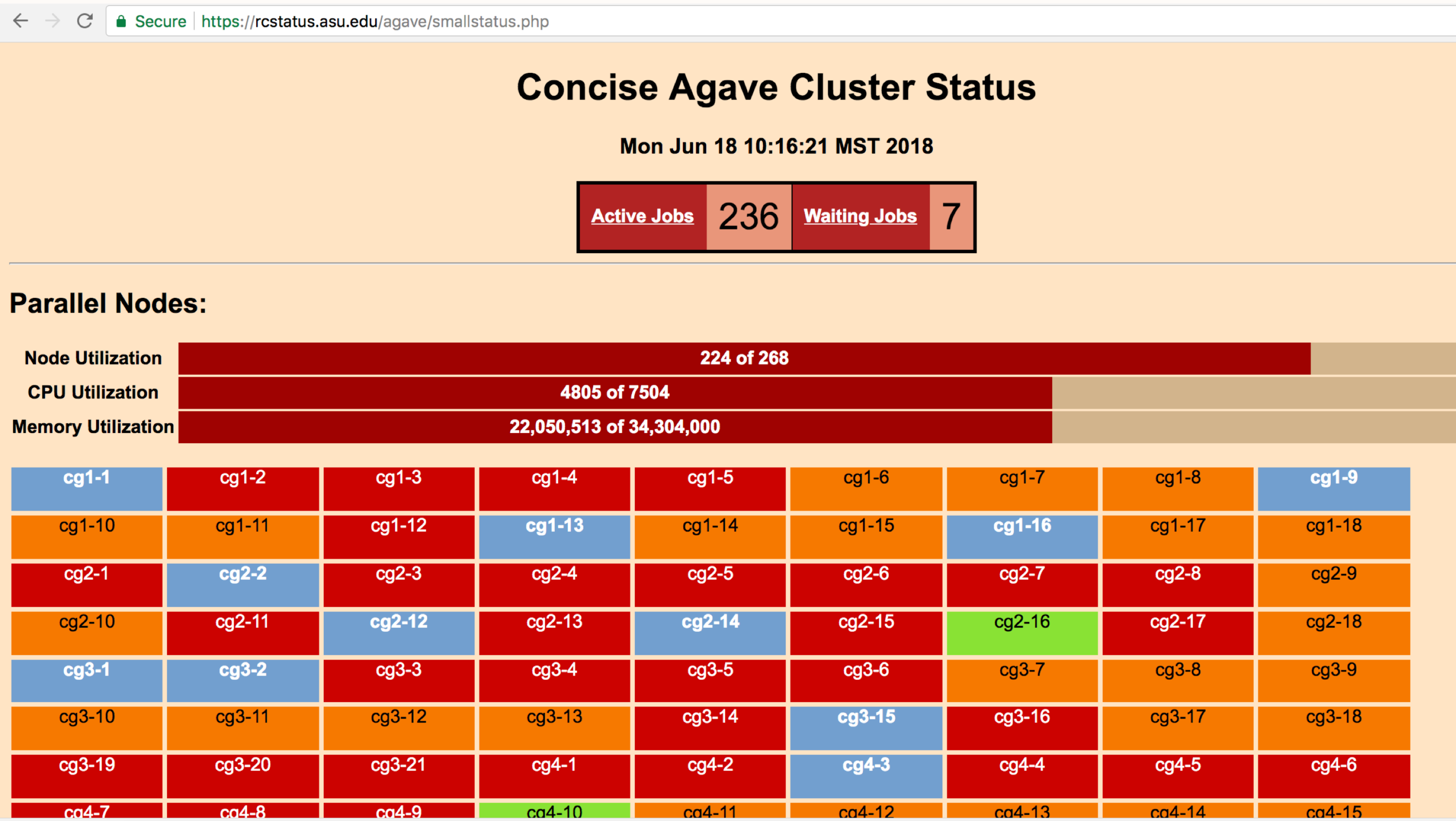
API: OpenMP

Distributed memory (MPP): each Processor has its own local memory. Must do message passing to exchange data between processors. (examples: Clusters)

Methods of memory access :
- various topological interconnects

API: Message Passing Interface (MPI)

Graphical View of Agave



Secure | https://rcstatus.asu.edu/agave/smallstatus.php

Concise Agave Cluster Status

Mon Jun 18 10:16:21 MST 2018

Active Jobs	236	Waiting Jobs	7

Parallel Nodes:

Node Utilization	224 of 268
CPU Utilization	4805 of 7504
Memory Utilization	22,050,513 of 34,304,000

cg1-1	cg1-2	cg1-3	cg1-4	cg1-5	cg1-6	cg1-7	cg1-8	cg1-9
cg1-10	cg1-11	cg1-12	cg1-13	cg1-14	cg1-15	cg1-16	cg1-17	cg1-18
cg2-1	cg2-2	cg2-3	cg2-4	cg2-5	cg2-6	cg2-7	cg2-8	cg2-9
cg2-10	cg2-11	cg2-12	cg2-13	cg2-14	cg2-15	cg2-16	cg2-17	cg2-18
cg3-1	cg3-2	cg3-3	cg3-4	cg3-5	cg3-6	cg3-7	cg3-8	cg3-9
cg3-10	cg3-11	cg3-12	cg3-13	cg3-14	cg3-15	cg3-16	cg3-17	cg3-18
cg3-19	cg3-20	cg3-21	cg4-1	cg4-2	cg4-3	cg4-4	cg4-5	cg4-6
cg4-7	cg4-8	cg4-9	cg4-10	cg4-11	cg4-12	cg4-13	cg4-14	cg4-15

Agave system Information

- Agave ~7.5K Broadwell cores
  - >300 trillion floating point operations per second (TFLOPs)
  - >34TB aggregate RAM
  - >6PB aggregate disk
  - Omnipath Interconnect – 100Gb/s
  - Located in ISTB1
- ~5K Xeon Phi cores
- GPU partition(s)
- Agave is a true *cluster* architecture
  - Each Broadwell node has 28 processors and 128GB of RAM (~4.5GB/CPU).
  - Programs needing more resources *must* use parallel programming
  - Normal, single processor applications *do not go faster* on Agave

Initial Login

- **From outside ASU campus, use VPN**: download from sslvpn.asu.edu
- Login with SSH or Putty (putty.org)

```
ssh ASURITE@agave.asu.edu
```

- Connects you to a login node
- Don't overwrite ~/.ssh/authorized_keys
  - Feel free to add to it if you know how to use it:
    - `ssh-keygen`
    - `ssh-copy-id -i ~/.ssh/id_rsa.pub ASURITE@agave.asu.edu`
  - SSH is used for job start up on the compute nodes. Mistakes can prevent your jobs from running
- For X forwarding, use `ssh –X`
- Nomachine: rcstatus.asu.edu/agave/howto
  - Click on "Logging in to Agave cluster with NoMachine Remote Desktop"
  - Get nomachine client and nomachine profile
- Xming, MobaXterm

Transferring files

- Secure copy

```
scp projectfile
 user1@agave.asu.edu:/home/user1/projectdir
scp –r projectdir
 user1@agave.asu.edu:/home/user1/projectdir
```

- `rsync -e ssh` for large file transfers
- `rsync –avtr bigfiledir user1@agave.asu.edu:/home/user1/projectdir`
- Winscp (winscp.net)
- ftp: Filezilla

Packages

- Modules are used to setup your PATH and other environment variables
- They are used to setup environments for packages & compilers

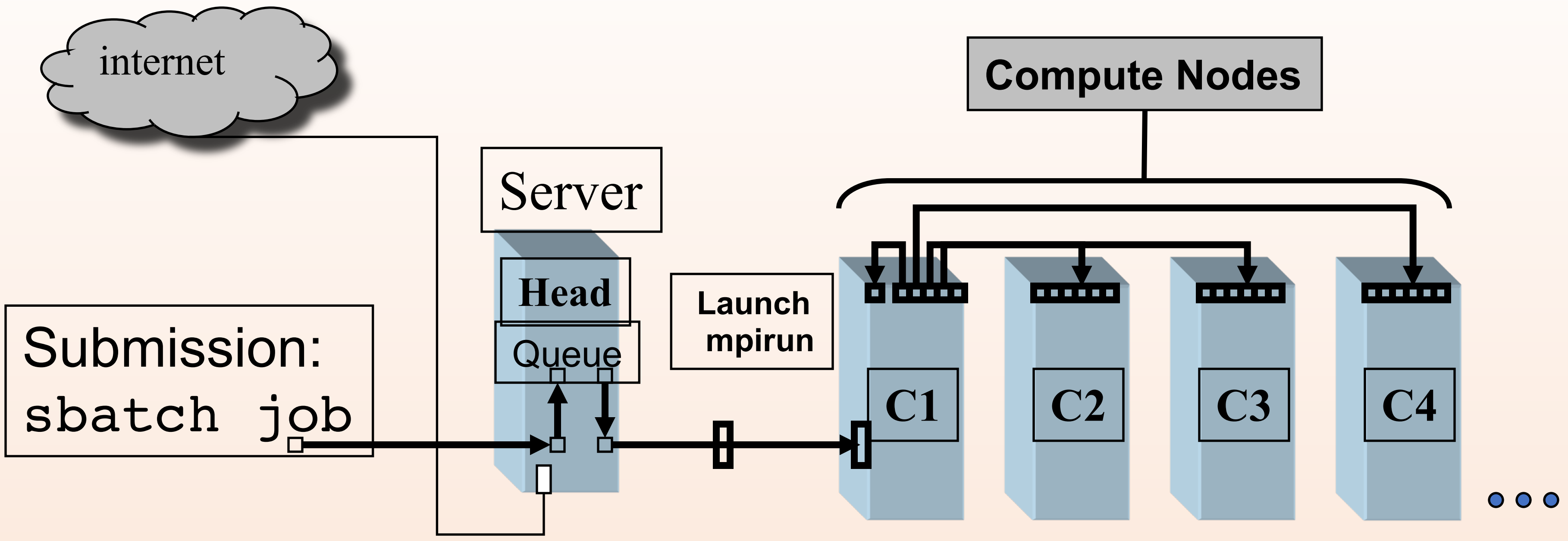
  `[user1@agave1 ~]$ module` {lists options}
  `[user1@agave1 ~]$ module avail` {lists available packages}
  `[user1@agave1 ~]$ module load <package> <…>` {add one or more packages}
  `[user1@agave1 ~]$ module unload <package>` {unload a package}

  `[user1@agave1 ~]$ module list` {lists loaded packages}

  `[user1@agave1 ~]$ module purge` {unloads all packages}

- Multiple compiler families available, so make sure you are consistent between libraries, source and run script!

Batch Submission Process



Queue: Job script waits for resources on Server
Compute nodes execute the job script, launching MPI processes

Launch: contact each compute node to start executable (e.g. a.out)

SLURM Commands

sbatch, srun, salloc	Submit a job
squeue	Check on the status of jobs
sinfo	Get info on nodes/partitions
scancel	Delete running and queued jobs
scontrol	Alter/hold/release jobs

man pages for all of these commands

showjobs	Running jobs with queue info
longjob <jobid>	Details on job

SLURM: OpenMP “job” Script

```
#!/bin/bash
#SBATCH -n 1                  }      # of cores
#SBATCH -J hello         }           Job name
#SBATCH -o %j.OUT  }                 stdout file name, %j = job id
#SBATCH -e %j.ERROR }                stderr file name, %j = job id
#SBATCH -t 0-00:15:00    }           Max Run Time (15 minutes)

module load intel/2018x
export OMP_NUM_THREADS=1       }     Execution commands
./hello
```

```
[user1@agave1 ~]$ sbatch job
```

SLURM: OpenMP Job Script II

```
#!/bin/bash
#SBATCH --nodes=2                          }  # of cores, architecture
#SBATCH --ntasks-per-node=28
#SBATCH --time=96:00:00                    }  walltime
#SBATCH --mail-type=ALL                    }  Send mail when job aborts/begins/ends
#SBATCH --mail-user=user1@asu.edu          }  Email address

module load openmpi/3.0.0-gnu-7x
export OMP_NUM_THREADS=28
export OMP_DYNAMIC=TRUE
time srun –mpi=pmi2 lmp_g++_openmpi –in in.eam –v x 2
```

Normal and wildfire queues

- The mybalance command:

```
[user1@agave1 ~]$ mybalance

User                    : user1
CPU Hours Allocated     : 25000
CPU Hours Used          : 526.36
CPU Hours Available     : 24473.63
```

- If the resource request for your submitted job (#CPUs X walltime requested) fits within your available CPU hours, the job is **non-preemptable**, i.e. will run uninterrupted in the **normal** queue.
- If your resource request exceeds the available CPU hours, the job is **preemptable**, and will run in the **wildfire** queue.
- Limit of 50 running jobs per user

SLURM partitions and environment

- Serial and parallel partitions
  - No need to specify – automatically determined based on requested resources
- `#SBATCH -p phi`
- `#SBATCH -p physicsgpu1 # Use physicsgpu1 partition`
- `#SBATCH -p cidsegpu1 # Use cidsegpu1 partition`

```
#SBATCH -q wildfire # Run job in wildfire QOS queue
#SBATCH --gres=gpu:2
```

Variable	Purpose
SLURM_JOB_ID	Batch job id
SLURM_SUBMIT_DIR	Directory where job was submitted
SLURM_JOB_NODELIST	Filename containing list of nodes
SLURM_ARRAY_TASK_ID	Slurm array index (next slide)

SLURM arrays and small jobs

- Arrays can loop to submit many jobs: `--array`
- `sbatch --array=0-20 job`

```
(in job) ./executable.x < $SLURM_ARRAY_TASK_ID.inp
 or ./executable2.x $SLURM_ARRAY_TASK_ID
```

- For single line short jobs: `--wrap`

```
 sbatch -n 2 --wrap="module load gcc/7x;gcc -fopenmp
hello_world.c;export OMP_NUM_THREADS=2;./a.out"
```

Matlab Job Script

```
#!/bin/bash
#SBATCH -n 1
#SBATCH -J hello
#SBATCH -o %j.OUT
#SBATCH -e %j.ERROR
#SBATCH -t 0-04:00:00

module load matlab/2018a
#Use one of the two commands below
matlab –nodisplay –nodesktop –nosplash < hello.m
matlab –nodisplay –nodesktop –nosplash –r “hello, quit”
```

# of cores

Job name

Output file name

Max Run Time (4 hours)

R Job Script

```
#!/bin/bash
#SBATCH -n 1                }      # of cores
#SBATCH -J hello            }      Job name
#SBATCH -o %j.OUT
#SBATCH -e %j.ERROR }              Output file name
#SBATCH -t 0-04:00:00   }          Max Run Time (4 hours)

module load r/3.5.1
#Use one of the two commands below
R --no-save --quiet --slave < regression.r
Rscript regression.r 20000
```

```
[user1@agave1 ~]$ sbatch job
```

NAMD “job” Script

```
#!/bin/bash
#SBATCH -N 1
#SBATCH -n 28
#SBATCH -J namd
#SBATCH -o %j.OUT
#SBATCH -e %j.ERROR
#SBATCH -t 1-00:00:00

module load namd/2.13-mpi
namd2 qwikmd_equilibration_0.conf
```

# of nodes

# of cores

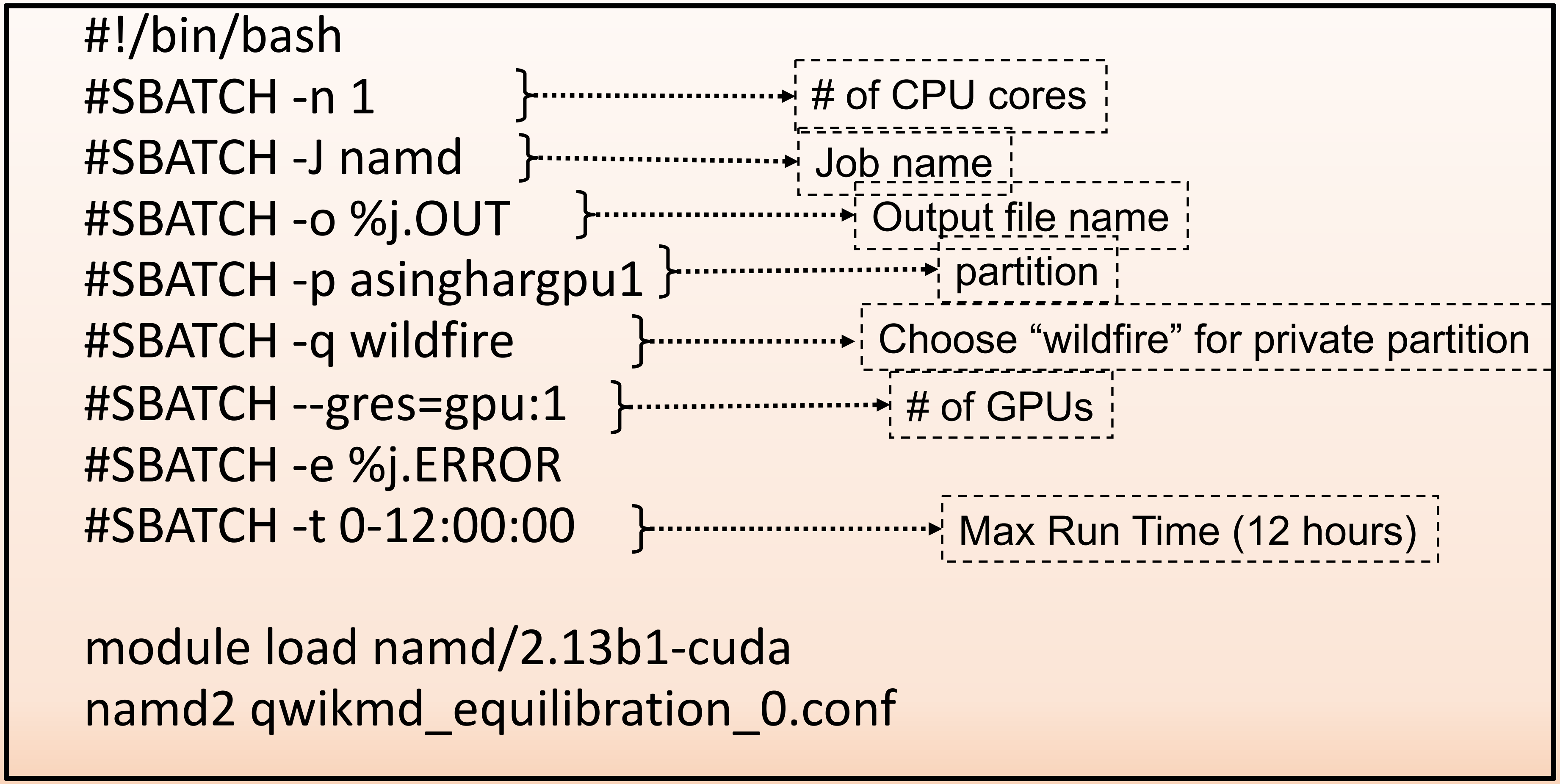
Job name

Output file name

Max Run Time (1 day)

```
[user1@agave1 ~]$ sbatch job
```

NAMD GPU "job" Script



```
#!/bin/bash
#SBATCH -n 1                 # of CPU cores
#SBATCH -J namd              Job name
#SBATCH -o %j.OUT            Output file name
#SBATCH -p asinghargpu1      partition
#SBATCH -q wildfire          Choose "wildfire" for private partition
#SBATCH --gres=gpu:1         # of GPUs
#SBATCH -e %j.ERROR
#SBATCH -t 0-12:00:00        Max Run Time (12 hours)

module load namd/2.13b1-cuda
namd2 qwikmd_equilibration_0.conf
```

```
[user1@agave1 ~]$ sbatch job
```

ASU Knowledge Enterprise Development
ARIZONA STATE UNIVERSITY

Job Monitoring (*squeue* utility)

```
[user1@agave1 ~]$ squeue
    JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
  208952_[0-199]    serial   COLD_3   mrline PD       0:00      1 (BeginTime)
           207709    serial R-px-OSa epopplet PD       0:00      1 (AssocMaxJobsLimit)
           207710    serial R-px-OSa epopplet PD       0:00      1 (AssocMaxJobsLimit)
           207711    serial R-px-OSa epopplet PD       0:00      1 (AssocMaxJobsLimit)
           207712    serial R-px-OSa epopplet PD       0:00      1 (AssocMaxJobsLimit)
           207713    serial R-px-OSa epopplet PD       0:00      1 (AssocMaxJobsLimit)
         .
         .
         .
```

Basic *squeue* options:

-u username	Display jobs belonging to specified user
-l, --long	Display extended job information

myjobs

To kill a job:

```
[user1@agave1 ~]$ scancel <JOBID>
```

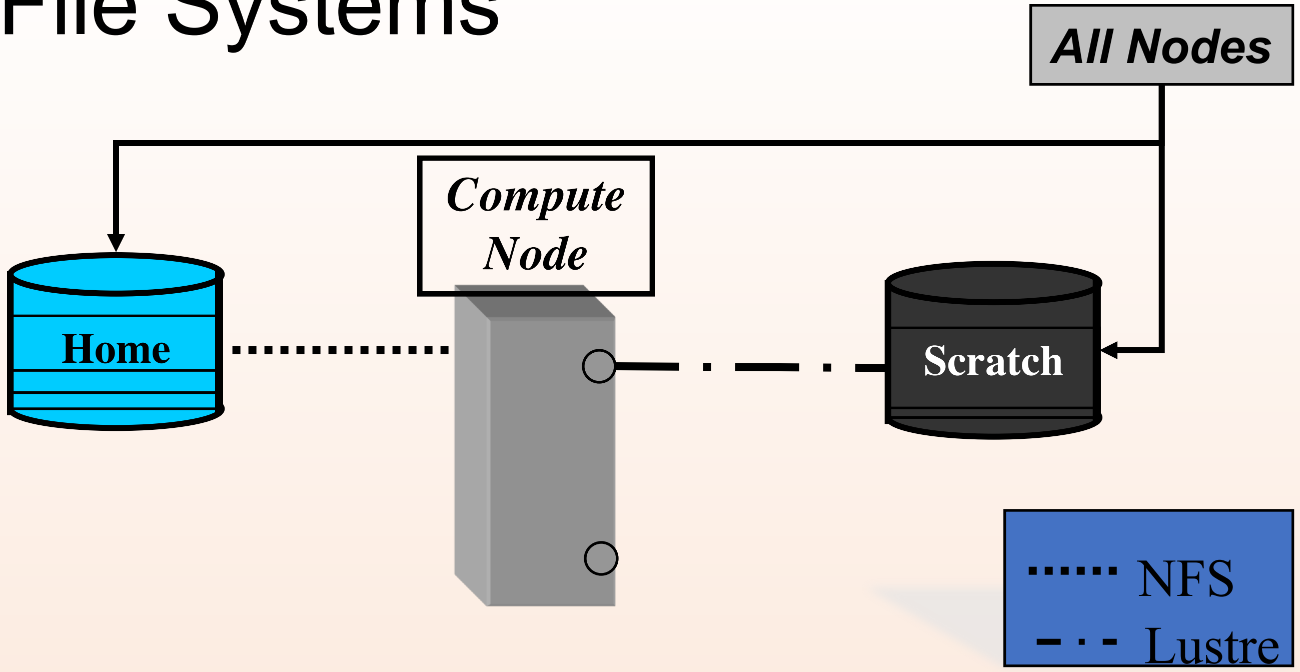
Interactive Mode

- `interactive` Interactive mode

- `screen` Detach “`^ad`” interactive jobs. Reattach with `screen -r`. (Also `tmux`)
  1. `screen` (on login node)
  2. `interactive`

- `interactive –N 1 –n 28` Entire Broadwell node
- `interactive –q wildfire –p asinghargpu1 --gres=gpu:1 –t4:00:00` GPU node

Available File Systems



Mount point	User Access Limit	Lifetime
/home	1TB quota	Project
/scratch	no quota	30 days

NSF sponsored resources: XSEDE, OSG

- **Stampede2 (TACC)**
  - **368280 CPUs**
  - **12.8 PFLOPs**
  - **Xeon Phi and Skylake**
- **Comet (SDSC)**
  - **46752 Haswell CPUs - 2 PFLOPs**
  - **Comet GPU – 104 TFLOPs**
- **Bridges (Pitt)**
  - **Bridges GPU – 900 TFLOPs**
  - **Bridges Large Memory nodes – 3TB and 12TB RAM nodes**
- **xsede.org – click on XUP, “create account”**
- **Open Science Grid – large-scale single node deploy**
- **Email `support@hpchelp.asu.edu` to reach ASU Campus Champions for assistance**

Good citizenship

- Shared login node: Do not compute on the login nodes
- Shared filesystem: Run I/O intensive jobs on scratch
- Shared network: Do not start 20 scps
- Shared compute resource: Give good estimate of runtime. Test submission scripts before submitting them at large scale.
- Shared help desk: Do some homework before submitting ticket. Do not submit multiple tickets on same topic. Describe issue in detail (e.g. job ID, full path to failing sbatch script, etc.). Be patient.

Conclusion

For any assistance please contact us:
`support@hpchelp.asu.edu`

Office hours: GWC546 1-4pm Tues (and 1-4pm Wed during academic year)

Info at: `researchcomputing.asu.edu` and `rcstatus.asu.edu/agave`