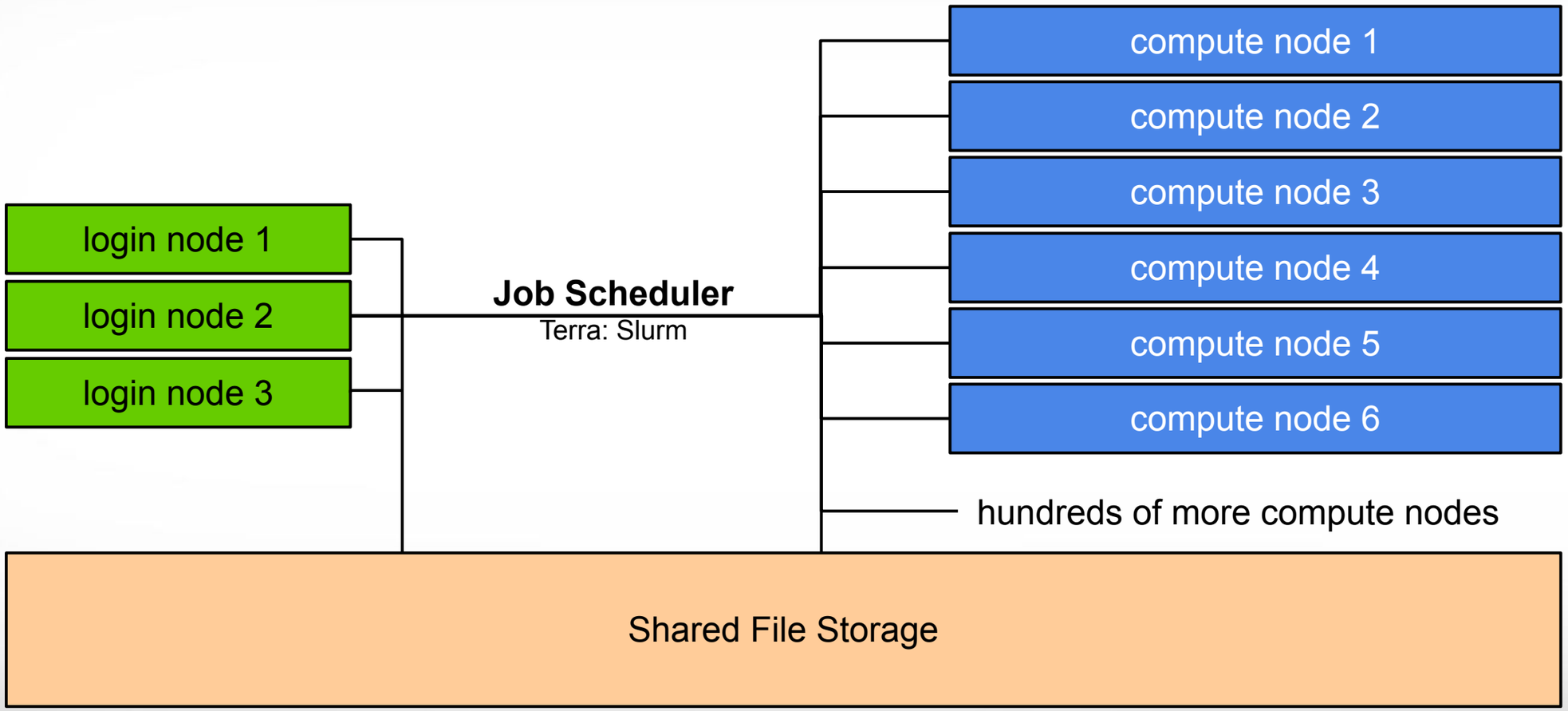
Terra

Slurm Job Scheduling Primer

HIGH PERFORMANCE RESEARCH COMPUTING
TEXAS A&M UNIVERSITY

Spring 2020

HPC Diagram


compute node 1
compute node 2
compute node 3
compute node 4
compute node 5
compute node 6
login node 1
login node 2
login node 3
Job Scheduler
Terra: Slurm
hundreds of more compute nodes
Shared File Storage

File Systems and User Directories

Directory	Environment Variable	Space Limit	File Limit	Intended Use
/home/$USER	$HOME	10 GB	10,000	Small to modest amounts of processing.
/scratch/user/$USER	$SCRATCH	1 TB	250,000	Temporary storage of large files for on-going computations. Not intended to be a long-term storage area.

- View usage and quota limits using the command: `showquota`
- Quota and file limit increases will only be considered for scratch and tiered directories
- Request a group directory for sharing files.

Batch Computing on HPRC Clusters

A batch job script is a text file that contains Unix and software commands and Batch manager job parameters

On-campus:

Campus Network

SSH

Login nodes

Create job

Job script file

Submit job

Batch manager:
LSF on Ada
Slurm on Terra

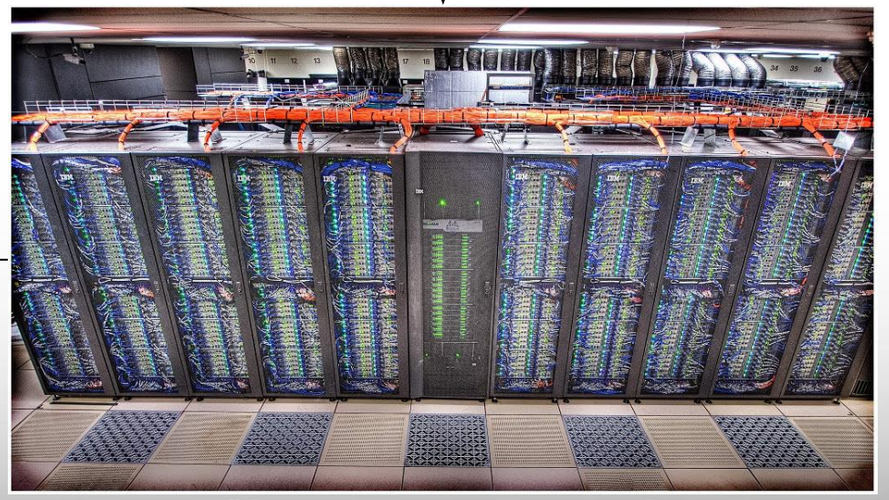
Queue

VPN

Internet

Off-campus:

Output Files



Cluster compute nodes

Sample Job Script Structure (Terra)

```
#!/bin/bash
##NECESSARY JOB SPECIFICATIONS
#SBATCH --export=NONE
#SBATCH --get-user-env=L
#SBATCH --job-name=JobExample1
#SBATCH --time=01:30:00
#SBATCH --ntasks=1
#SBATCH --mem=2G
#SBATCH --output=stdout.%j

##OPTIONAL JOB SPECIFICATIONS
#SBATCH --account=123456
#SBATCH --mail-type=ALL
#SBATCH --mail-user=email_address

# load required module(s)
module load Python/3.7.0-intel-2018b

./my_program.py
```

These parameters describe your job to the job scheduler

This is single line comment and not run as part of the script

Load the required module(s) first

This is a command that is executed by the job

Important Batch Job Parameters

Terra	Ada	Comment
`#SBATCH --export=NONE` `#SBATCH --get-user-env=L`	`#BSUB -L /bin/bash`	Initialize job environment.

`#SBATCH --time=HH:MM:SS`	`#BSUB -W HH:MM`
	`or`
	`#BSUB -W MM`
		Specifies the time limit for the job.
		Must specify seconds SS on Terra

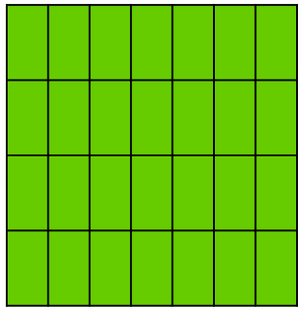
`#SBATCH --ntasks=NNN`	`#BSUB -n NNN`	Total number of tasks (cores) for the job.
`#SBATCH --ntasks-per-node=XX`	`#BSUB -R "span[ptile=XX]"`	Specifies the maximum number of tasks (cores) to allocate per node

`#SBATCH --mem=nnnnM`
`or`
`#SBATCH --mem=nG`
`(memory per NODE)`
	`#BSUB -R "rusage[mem=nnnn]"`
	`#BSUB -M nnnn`
	`(memory per CORE)`
		Sets the maximum amount of memory (MB).
		G for GB is supported on Terra

hprc.tamu.edu/wiki/HPRC:Batch_Translation

Mapping Jobs to Cores per Node on Terra

A.

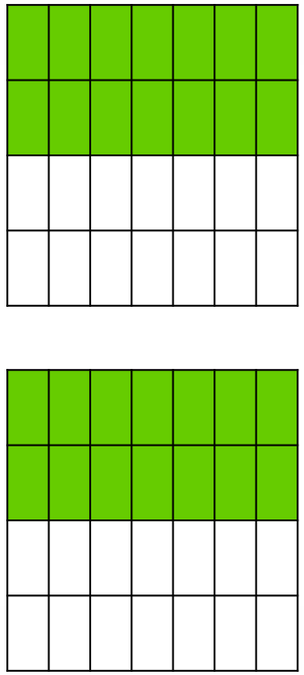


28 cores on
1 compute node

#SBATCH --ntasks 28
#SBATCH --tasks-per-node=28

Preferred Mapping
(if applicable)

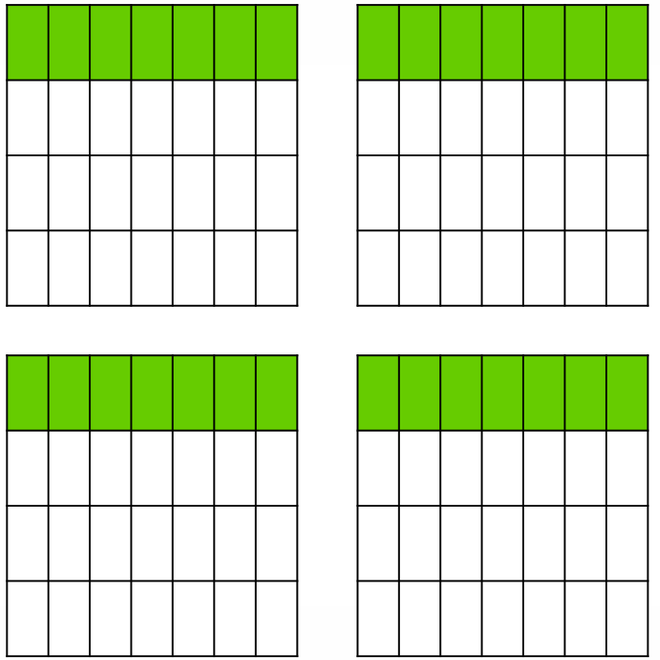
B.



28 cores on
2 compute nodes

#SBATCH --ntasks 28
#SBATCH --tasks-per-node=14

C.



28 cores on
4 compute nodes

#SBATCH --ntasks 28
#SBATCH --tasks-per-node=7

Job Memory Requests on **Terra**

- Specify memory request based on memory per node:

  #SBATCH --mem=xxxxM # memory per node in MB

  or

  #SBATCH --mem=xG # memory per node in GB

- On 64GB nodes, usable memory is at most 56 GB. The per-process memory limit should not exceed 2000 MB for a 28-core job.
- On 128GB nodes, usable memory is at most 112 GB. The per-process memory limit should not exceed 4000 MB for a 28-core job.

Consumable Computing Resources

- Resources specified in a job file:
  - Processor cores
  - Memory
  - Wall time
  - GPU
- Service Unit (SU) - Billing Account
  - Use "**myproject**" to query
    hprc.tamu.edu/wiki/HPRC:AMS:Service_Unit

- Other resources:
  - Software license/token
    - Use "**license_status**" to query
    - hprc.tamu.edu/wiki/SW:License_Checker

myproject

```
================================================================================
             List of YourNetID's Project Accounts
--------------------------------------------------------------------------------
|   Account    |   FY   | Default | Allocation |Used & Pending SUs|   Balance   |    PI     |
--------------------------------------------------------------------------------
|1228000223136|  2019 |        N|    10000.00|              0.00|    10000.00|Doe, John  |
--------------------------------------------------------------------------------
|1428000243716|  2019 |        Y|     5000.00|           -71.06|     4928.94|Doe, Jane  |
--------------------------------------------------------------------------------
```

Find available license for "ansys":

license_status -s ansys

```
License status for ANSYS:
----------------------------------------------------------------
|License Name                |  # Issued|  # In Use|# Available|
----------------------------------------------------------------
|aa_mcad                     |        50|         0|         50|
|aa_r                        |        50|        32|         18|
|aim_mp1                     |        50|         0|         50|
|    ......                                                    |
----------------------------------------------------------------
```

Find detail options:

license_status -h

Terra: Examples of SUs charged based on Job Cores, Time and Memory Requested

A Service Unit (SU) on **Terra** is equivalent to one core or **2** GB memory usage for one hour.

	Number of Cores	GB of memory per core	Total Memory (GB)	Hours	SUs charged
1.	1	2	2	1	1
2.	1	3	3	1	2
3.	1	56	56	1	28
4.	28	2	56	1	28

hprc.tamu.edu/wiki/HPRC:AMS:Service_Unit

Batch Queues

- Job submissions are auto-assigned to batch queues based on the resources requested (number of cores/nodes and walltime limit)
- Some jobs can be directly submitted to a queue:
  - On **Terra**, if gpu nodes are needed, use the gpu partition/queue:
    ```
    #SBATCH --partition=gpu
    ```
  - Jobs that have special resource requirements are scheduled in the special queue (must request access to use this queue)

hprc.tamu.edu/wiki/Terra:Batch#Queues

sinfo : Current Queues on **Terra**

```
File   Edit   View   Search   Terminal   Help
[   netid  @terra2 ~]$ sinfo
PARTITION        AVAIL  TIMELIMIT    JOB_SIZE    NODES(A/I/O/T)   CPUS(A/I/O/T)
short*           up     2:00:00      1-16        156/145/3/304    3667/4761/84/8512
medium           up     1-00:00:00   1-64        156/145/3/304    3667/4761/84/8512
long             up     7-00:00:00   1-32        156/145/3/304    3667/4761/84/8512
gpu              up     2-00:00:00   1-48        48/0/0/48        797/547/0/1344
vnc              up     12:00:00     1           48/0/0/48        797/547/0/1344
xlong            up     21-00:00:00  1-32        108/145/3/256    2870/4214/84/7168
staff            up     infinite     1-infinite  156/145/3/304    3667/4761/84/8512
low_priority     up     1-00:00:00   1-infinite  156/145/3/304    3667/4761/84/8512
special          up     7-00:00:00   1-infinite  156/145/3/304    3667/4761/84/8512
knl              up     7-00:00:00   1-8         0/14/2/16        0/980/140/1120
```

For the NODES and CPUS columns:
- **A = Active (in use by running jobs)**
- **I = Idle (available for jobs)**
- **O = Offline (unavailable for jobs)**
- **T = Total**

Queue Limits on Terra

Queue	Job Max Cores / Nodes	Job Max Walltime	Compute Node Types	Per-User Limits Across Queues	Notes
short	448 cores / 16 nodes	2 hrs	64 GB nodes (256) 128 GB nodes with GPUs (36)	1800 cores per user	
medium	1792 cores / 64 nodes	1 day			
long	896 cores / 32 nodes	7 days			
xlong	448 cores / 16 nodes	21 days	64 GB nodes (256)	448 cores per user	--partition xlong
gpu	1344 cores / 48 nodes	2 days	128 GB nodes with GPUs (48)		For jobs requiring GPUs.
vnc	28 cores / 1 node	12 hours	128 GB nodes with GPUs (48)		For remote visualization jobs
knl	68 cores / 8 nodes 72 cores / 8 nodes	7 days	96 GB nodes with KNL processors (16)		For jobs requiring a KNL processor

Submitting Your Job and Check Job Status

Submit job

```
sbatch example01.job
```

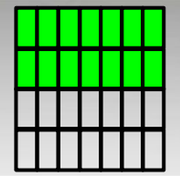
```
Submitted batch job 161997
(from job_submit) your job is charged as below
                Project Account: 122792016265
                Account Balance: 1687.066160
                Requested SUs:   3
```

Check status

```
squeue -u netID
```

```
JOBID   NAME      USER      PARTITION  NODES  CPUS  STATE    TIME   TIME_LEFT  START_TIME          REASON      NODELIST
64039   somejob   someuser  medium     4      112   PENDING  0:00   20:00      2017-01-30T21:00:4  Resources
64038   somejob   someuser  medium     4      112   RUNNING  2:49   17:11      2017-01-30T20:40:4  None        tnxt-[0401-0404]
```

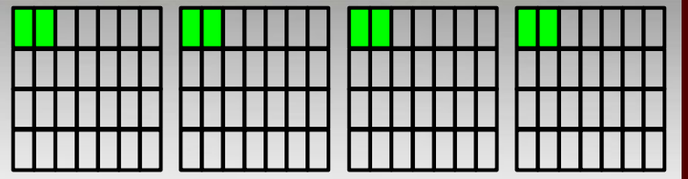
Terra Job File (multi core, single node)



```
#!/bin/bash
##ENVIRONMENT SETTINGS; CHANGE WITH CAUTION
#SBATCH --export=NONE                # Do not propagate environment
#SBATCH --get-user-env=L             # Replicate login environment

##NECESSARY JOB SPECIFICATIONS
#SBATCH --job-name=JobExample2       # Set the job name to "JobExample2"
#SBATCH --time=6:30:00               # Set the wall clock limit to 6hr and 30min
#SBATCH --nodes=1                    # Request 1 node
#SBATCH --ntasks-per-node=14         # Request 14 tasks(cores) per node
#SBATCH --mem=28G                    # Request 28GB per node
#SBATCH --output=stdout.%j           # Send stdout to "stdout.[jobID]"
#SBATCH --error=stderr.%j            # Send stderr to "stderr.[jobID]"
##OPTIONAL JOB SPECIFICATIONS
#SBATCH --account=123456     # Set billing account to 123456  #find your account with "myproject"
#SBATCH --mail-type=ALL              # Send email on all job events
#SBATCH --mail-user=email_address    # Send all emails to email_address

# load required module(s)
module load intel/2018b

# run your program
./my_multicore_program
```

SUs = 91

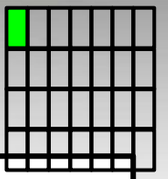
Terra Job File (multi core, multi node)



```
#!/bin/bash
##ENVIRONMENT SETTINGS; CHANGE WITH CAUTION
#SBATCH --export=NONE                # Do not propagate environment
#SBATCH --get-user-env=L             # Replicate login environment

##NECESSARY JOB SPECIFICATIONS
#SBATCH --job-name=JobExample3       # Set the job name to "JobExample3"
#SBATCH --time=1-12:00:00            # Set the wall clock limit to 1 Day and 12hr
#SBATCH --ntasks=8                   # Request 8 tasks (cores)
#SBATCH --ntasks-per-node=2          # Request 2 tasks(cores) per node
#SBATCH --mem=2.5G                   # Request 2.5 GB  per node
#SBATCH --output=stdout.%j           # Send stdout and stderr to "stdout.[jobID]"

##OPTIONAL JOB SPECIFICATIONS
#SBATCH --account=123456             # Set billing account to 123456 #find your account with "myproject"
#SBATCH --mail-type=ALL              # Send email on all job events
#SBATCH --mail-user=email_address    # Send all emails to email_address

# this intel toolchain is just an example.  recommended toolchain is TBD
module load intel/2017A

# run program with MPI
mpirun my_multicore_multinode_program
```

SUs = 288

Terra Job File (serial GPU)



```
#!/bin/bash
##ENVIRONMENT SETTINGS; CHANGE WITH CAUTION
#SBATCH --export=NONE                # Do not propagate environment
#SBATCH --get-user-env=L             # Replicate login environment

##NECESSARY JOB SPECIFICATIONS
#SBATCH --job-name=JobExample4       # Set the job name to "JobExample4"
#SBATCH --time=01:00:00              # Set the wall clock limit to 1hr
#SBATCH --ntasks=1                   # Request 1 task (core)
#SBATCH --mem=2G                     # Request 2GB per node
#SBATCH --output=stdout.%j           # Send stdout and stderr to "stdout.[jobID]"
#SBATCH --gres=gpu:1                 # Request 1 GPU
#SBATCH --partition=gpu              # Request the GPU partition/queue

##OPTIONAL JOB SPECIFICATIONS
#SBATCH --account=123456          # Set billing account to 123456  #find your account with "myproject"
#SBATCH --mail-type=ALL              # Send email on all job events
#SBATCH --mail-user=email_address    # Send all emails to email_address

# load required module(s)
module load intel/2017A
module load CUDA/9.2.148.1

# run your program
./my_gpu_program
```

SUs = 28

Other Type of Jobs

- MPI and OpenMP
- Visualization:
  - portal.hprc.tamu.edu (visualization jobs can be run on both Ada and Terra; more details in later slide)
- Large number of concurrent single core jobs
  - Check out ***tamulauncher***
    - hprc.tamu.edu/wiki/SW:tamulauncher
    - Useful for running many single core commands concurrently across multiple nodes within a job
    - Can be used with serial or multi-threaded programs
    - Distributes a set of commands from an input file to run on the cores assigned to a job
    - Can only be used in batch jobs
    - If a tamulauncher job gets killed, you can resubmit the same job to complete the unfinished commands in the input file

Job Submission and Tracking

Terra	Description
`sbatch jobfile1`	Submit jobfile1 to batch system
`squeue [-u user_name] [-j job_id]`	List jobs
`scancel job_id`	Kill a job
`sacct -X -j job_id`	Show information for a job (can be when job is running or recently finished)
`sacct -X -S YYYY-HH-MM`	Show information for all of your jobs since YYYY-HH-MM
`lnu job_id`	Show resource usage for a job
`pestat -u $USER`	Show resource usage for a running job
`seff job_id`	Check CPU/memory efficiency for a job

hprc.tamu.edu/wiki/HPRC:Batch_Translation

Check your Service Unit (SU) Balance

- List the SU Balance of your Account(s)

```
myproject
```

```
 ================================================================================
              List of YourNetID's Project Accounts
--------------------------------------------------------------------------------
|   Account    |    FY    | Default | Allocation |Used & Pending SUs|   Balance  |    PI    |
--------------------------------------------------------------------------------
|1228000223136|  2019    |       N|    10000.00|              0.00|    10000.00|Doe, John |
--------------------------------------------------------------------------------
|1428000243716|  2019    |       Y|     5000.00|            -71.06|     4928.94|Doe, Jane |
--------------------------------------------------------------------------------
|1258000247058|  2019    |       N|     5000.00|             -0.91|     4999.09|Doe, Jane |
--------------------------------------------------------------------------------
```

- To specify a project ID to charge in the job file
  - Terra: **`#SBATCH -A Account#`**

- Run "**`myproject -d Account#`**" to change default project account
- Run "**`myproject -h`**" to see more options

hprc.tamu.edu/wiki/HPRC:AMS:Service_Unit
hprc.tamu.edu/wiki/HPRC:AMS:UI

Job submission issue: insufficient SUs

Terra:

```
$ sbatch myjob
sbatch: error: (from job_submit) your account's balance is not sufficient to submit your job
               Project Account: 123940134739
               Account Balance: 382.803877
               Requested SUs:   18218.666666667
```

- What to do if you need more SUs
  - Ask your PI to transfer SUs to your account
  - Apply for more SUs (if you are eligible, as a PI or permanent researcher)

hprc.tamu.edu/wiki/HPRC:CommonProblems#Q:_How_do_I_get_more_SUs.3F
hprc.tamu.edu/wiki/HPRC:AMS:Service_Unit
hprc.tamu.edu/wiki/HPRC:AMS:UI

List Node Utilization on **Terra**: *lnu*

```
lnu jobid
```

lists the node utilization across all nodes for a running job.
to see more options use: `lnu -h`

Example:

```
lnu 565849
```

Note: Slurm updates the node information every few minutes

```
JOBID    NAME       USER       PARTITION    NODES    CPUS     STATE        TIME     TIME_LEFT        START_TIME
565849   somename   someuser   long         3        84       RUNNING      17:37    6-23:42:23       2018-01-25T15:19:55

HOSTNAMES     CPU_LOAD    FREE_MEM    MEMORY   CPUS(A/I/O/T)
tnxt-0703     26.99       53462       57344    28/0/0/28
tnxt-0704     26.93       52361       57344    28/0/0/28
tnxt-0705     26.95       47166       57344    28/0/0/28
```

Note: CPU_LOAD is not the same as % utilization

For the CPUS columns:
- **A = Active (in use by running jobs)**
- **I = Idle (available for jobs)**
- **O = Offline (unavailable for jobs)**
- **T = Total**

Monitor Compute Node Utilization on **Terra**: *pestat*

`pestat [-u username]`

lists the node utilization across all nodes for a running job.
to see more options use: `pestat -h`

Example:

```
pestat -u $USER
```

Hostname	Partition	Node State	Num_CPU Use/Tot	CPUload	Memsize (MB)	Freemem (MB)	Joblist JobId User ...
tnxt-0703	xlong	alloc	28	28	16.23*	57344	55506	565849 someuser
tnxt-0704	xlong	alloc	28	28	19.60*	57344	53408	565849 someuser
tnxt-0705	xlong	alloc	28	28	19.56*	57344	53408	565849 someuser

Low CPU load utilization highlighted in **Red**
(Freemem should also be noted)

```
pestat -u $USER
```

Hostname	Partition	Node State	Num_CPU Use/Tot	CPUload	Memsize (MB)	Freemem (MB)	Joblist JobId User ...
tnxt-0703	xlong	alloc	28	28	27.54	57344	55506	565849 someuser
tnxt-0704	xlong	alloc	28	28	27.50	57344	53408	565849 someuser
tnxt-0705	xlong	alloc	28	28	26.47*	57344	53408	565849 someuser

Good CPU load utilization highlighted in **Purple**
Ideal CPU load utilization displayed in White

Job Environment Variables

- **Terra:**
  - **$SLURM_JOBID** = job id
  - **$SLURM_SUBMIT_DIR** = directory where job was submitted from
  - **$SCRATCH** = /scratch/user/NetID
  - **$TMPDIR** = /work/job.$SLURM_JOBID
    - $TMPDIR is local to each assigned compute node for the job and is about 850GB

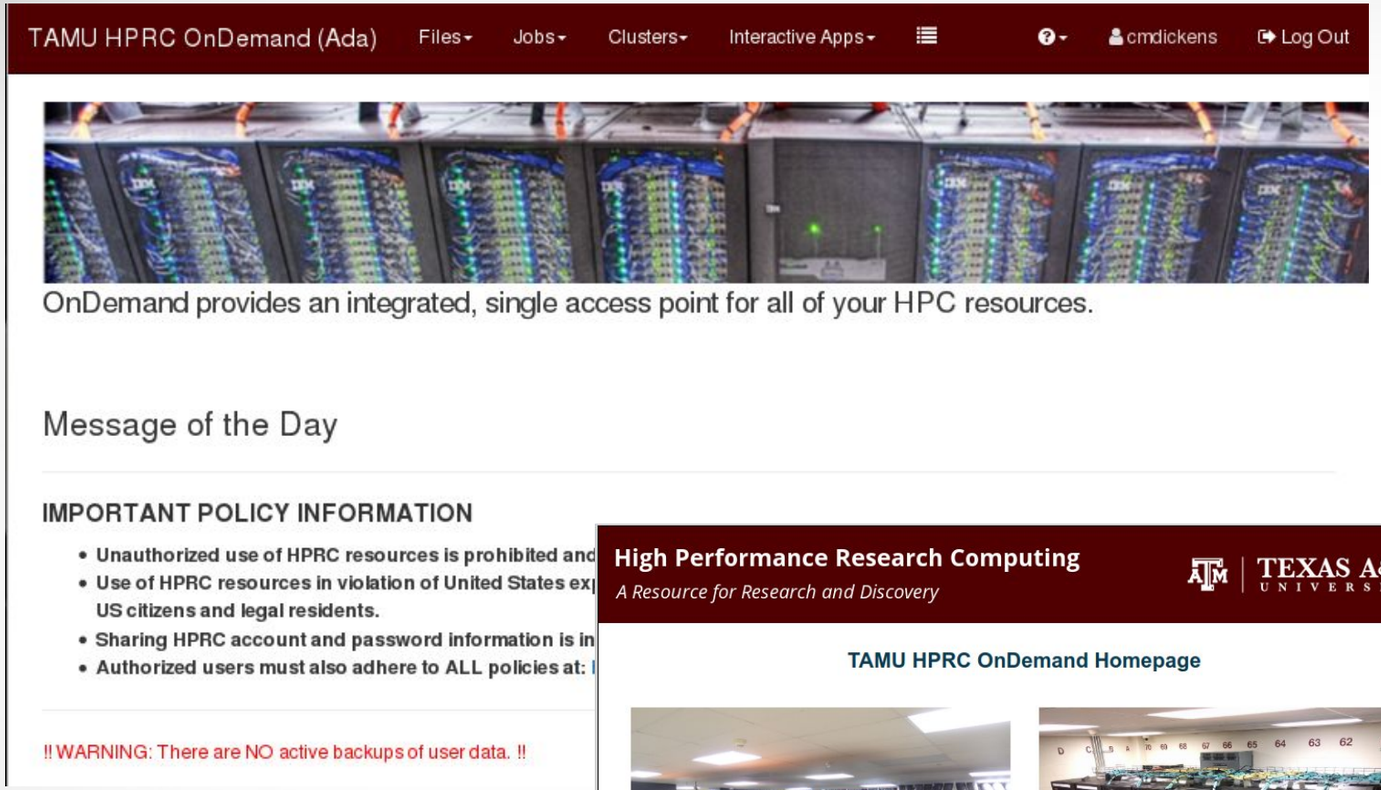
hprc.tamu.edu/wiki/Ada:Batch_Processing_LSF#Environment_Variables
hprc.tamu.edu/wiki/Terra:Batch#Environment_Variables

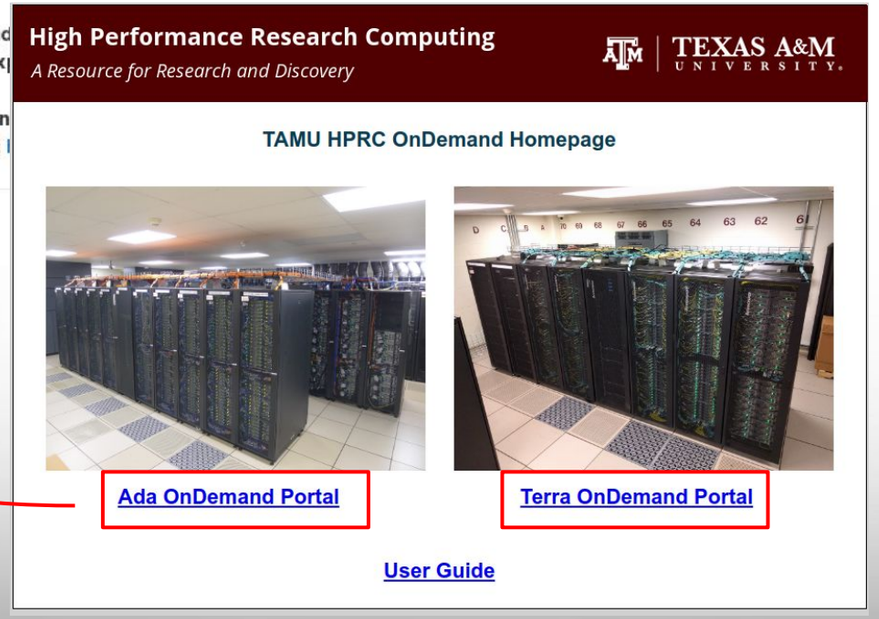
Jobs using tamubatch

Automatic batch job script that submits jobs for the user without the need of writing a full batch script on the cluster

Access help with: `tamubatch --help`

portal.hprc.tamu.edu





https://hprc.tamu.edu/wiki/SW:Portal

The HPRC portal allows users to do the following

- Browse files on the filesystem
- Access the Ada, Terra, Curie Unix command line
- Launch jobs
- Compose job scripts
- Launch interactive GUI apps (SUs charged)

HIGH PERFORMANCE RESEARCH COMPUTING

TEXAS A&M UNIVERSITY

Thank you.

Any questions?