HPRC Clusters:

Terra Primer

Windows Users:

To follow along, please download Mobaxterm
(a free SSH client)

Google Search “Mobaxterm” or navigate to:
https://mobaxterm.mobatek.net/download.html

Download the “Installer edition” (green button)

MobaXterm Home Edition v20.2
(Installer edition)

HIGH PERFORMANCE RESEARCH COMPUTING
TEXAS A&M UNIVERSITY

Spring 2022

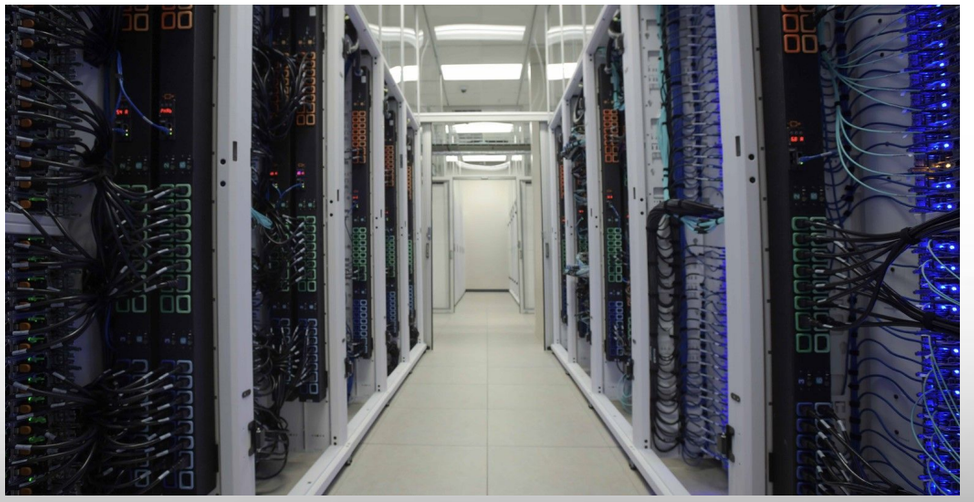
High Performance Research Computing Resources

Terra

320-node hybrid Intel cluster from Lenovo with an Omni-Path Architecture (OPA) interconnect and 48 NVIDIA K80 dual-GPU accelerators. 304 nodes based on the Intel Broadwell processor & 16 nodes based on the Intel Knights Landing processor. 4 nodes with Skylake processors, 192 GB of memory, and dual V100 GPUs.

Grace

925-node Intel cluster from Dell with an InfiniBand HDR-100 interconnect, A100 GPUs, RTX 6000 GPUs and T4 GPUs. All nodes are based on the Intel Cascade Lake processor.



High Performance Research Computing Resources

FASTER

184-node Intel cluster from Dell with an InfiniBand HDR-100 interconnect. A100 GPUs, A10 GPUs, A30 GPUs, A40 GPUs and T4 GPUs are distributed and composable via Liqid PCIe fabrics. All nodes are based on the Intel Ice Lake processor.

ViDaL

24-node Dell cluster with Intel Skylake processors and a 40Gb Ethernet interconnect. ViDaL provides secure and compliant computing facilities to conduct research projects involving analysis of sensitive or proprietary data. ViDaL also offers support for both Windows and Linux operating systems.

HPRC Clusters

Terra
28 cores/node

Login Nodes	3
64 GB memory compute nodes	256
KNL (Knights Landing) (no SUs charged)	16
K80 GPU 128 GB memory compute nodes	48
V100 GPU 192 GB compute nodes	4

hprc.tamu.edu/resources

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:

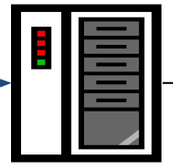
Campus Network

VPN

Internet

Off-campus:

SSH



Login nodes

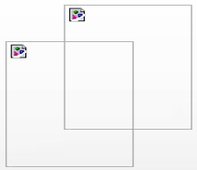
Create job

Job script file

Submit job

Batch manager:
Slurm on Terra

Queue



Output Files

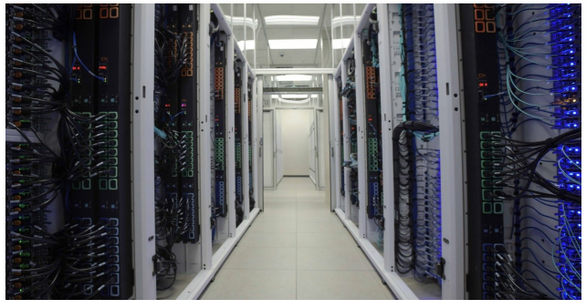
Cluster compute nodes

Accessing Terra

- SSH command is required for accessing Grace / Terra:
  - On campus: `ssh NetID@terra.tamu.edu`
  - Off campus:
    - Set up and start VPN (Virtual Private Network): u.tamu.edu/VPnetwork
    - Then: `ssh NetID@terra.tamu.edu`
  - *Two-Factor Authentication* enabled for CAS, VPN, SSH (see next slide for more detail)
- SSH programs for Windows:
  - MobaXTerm (preferred, includes SSH and X11)
  - PuTTY SSH
- Access through portal.hprc.tamu.edu (Menu “Clusters” => “Terra Shell Access”)
- Terra has 3 login nodes. Check the bash prompt.
  - `[NetID@terra3 ~]$`
- Login sessions that are idle for **60** minutes will be closed automatically
- Processes run longer than **60** minutes on login nodes will be killed automatically.
- **Do not use more than 8 cores on the login nodes!**
- **Do not use the sudo command.**

hprc.tamu.edu/wiki/HPRC:Access

Pop Quiz



Which one of the following is not an HPRC cluster?

A. FASTER
B. Bozo
C. Grace
D. Terra

Two-Factor Authentication

- Duo NetID two-factor authentication to enhance security (it.tamu.edu/duo/)
  - All web login (howdy, portal.hprc.tamu.edu, Globus) through CAS
  - VPN to TAMU campus (since Oct 1st, 2018)
  - SSH/SFTP to HPRC clusters (since Nov 4th, 2019)

- See instructions in two-factor wiki page (hprc.tamu.edu/wiki/Two_Factor)
- SSH clients work with Duo
  - ssh command from Linux, macOS Terminal, Windows cmd
  - MobaXterm for Windows (click on “Session” icon or via local session: hit “enter” 3 times and wait for “Password:” prompt)
  - Putty for Windows
- SFTP clients work with Duo
  - scp/sftp command from Linux, macOS Terminal, Windows cmd
  - WinSCP for Windows
  - Cyberduck for macOS

- Not all software supports SSH+Duo: SFTP in Matlab

hprc.tamu.edu/wiki/Two_Factor

Example: SSH login with Duo

```
$ ssh terra.tamu.edu
**********************************************************************
…. warning message (snipped) …….
**********************************************************************

Password:
Duo two-factor login for UserNetID

Enter a passcode or select one of the following options:

 1. Duo Push to XXX-XXX-1234
 2. Phone call to XXX-XXX-1234
 3. SMS passcodes to XXX-XXX-1234 (next code starts with: 9)

Passcode or option (1-3): 1
Success. Logging you in...
```

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.

- **`$HOME`** and **`$SCRATCH`** directories are not shared between Grace and Terra clusters.
- 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.
- Do not share your home or scratch directories.

hprc.tamu.edu/wiki/Terra:Filesystems_and_Files

Software

- See the Software wiki page for instructions and examples
  - hprc.tamu.edu/wiki/SW
  - hprc.tamu.edu/software/terra

- License-restricted software
  - Contact license owner for approval
- Contact us for software installation help/request
  - User can install software in their home/scratch dir
  - Do not run the "*sudo*" command when installing software

Application Modules

- Installed applications are available as modules which are available to all users
- Purge unused modules before loading new modules.
- **Avoid loading modules in your `~/.bashrc`**

```
module list                          # list all loaded modules

module spider matlab                 # search for matlab

module load Matlab/R2019b            # load matlab R2019b

module list                          # list all loaded modules

module purge                         # remove all loaded modules
```

Modules and Toolchains

- Load modules with the same toolchains in your job scripts
- The **`2018b`** and **`GCCcore-7.3.0`** toolchain versions are recommended
    - `intel/2018b`
    - `iomkl/2018b`
    - `foss/2018b`
    - `GCCcore/7.3.0`

- Do not mix toolchains in the same job script

```
module load HISAT2/2.0.4-foss-2016b
module load TopHat/2.1.1-intel-2017A-Python-2.7.12
module load Cufflinks/2.2.1-intel-2015B
```

- Same rule applies to compilers and libraries.

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

`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   |
--------------------------------------------------------------------------------
```

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

`license_status -a`

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`

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 |
--------------------------------------------------------------------------------
```

- 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

Batch Queues

Job submissions are auto-assigned to batch queues based on the resources requested (number of cores/nodes and walltime limit)

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

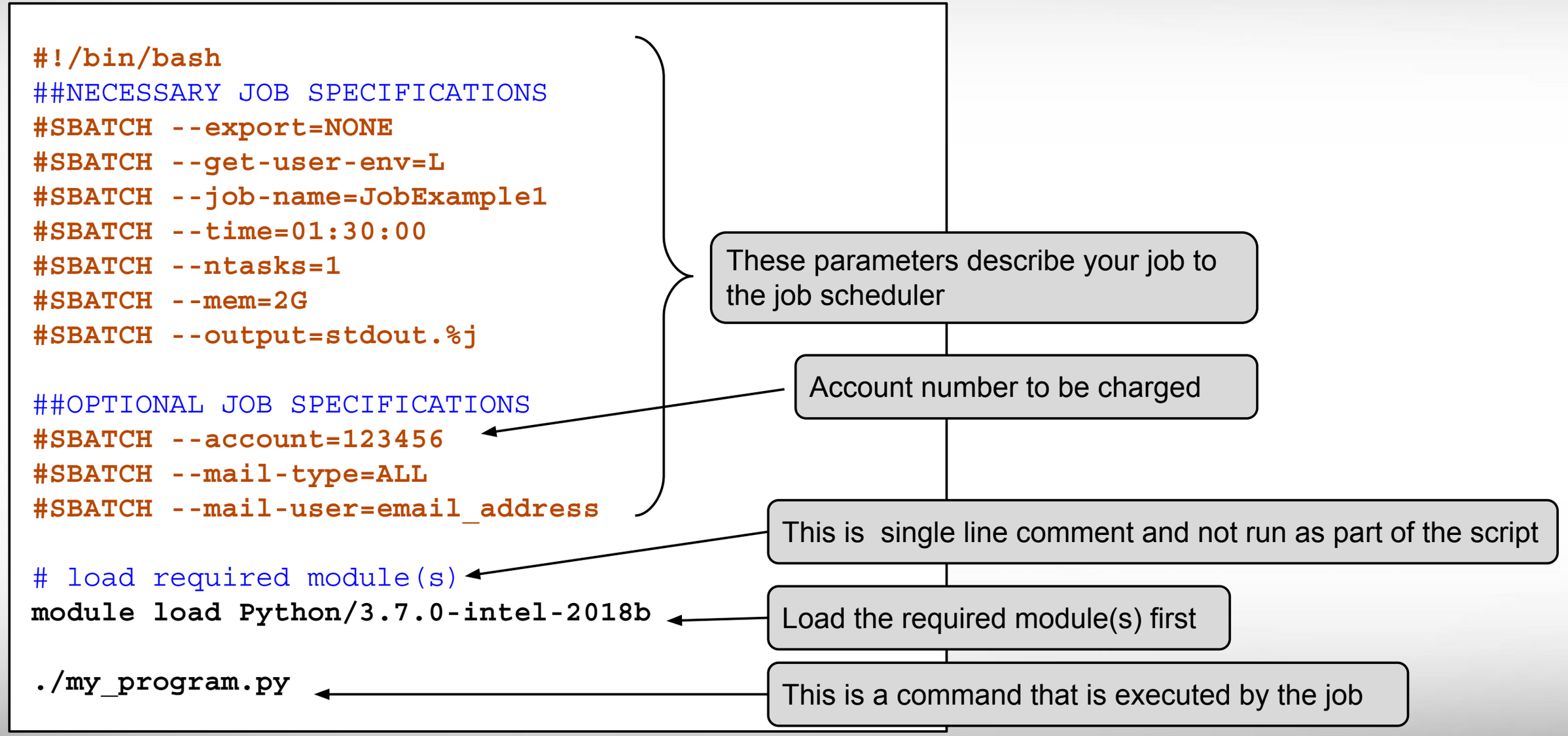
sinfo : Current Queues on **Terra**

```
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**

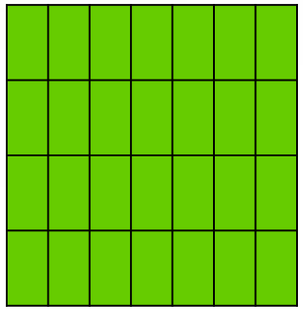
Sample Job Script Structure (Terra)

```
#!/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
```



Mapping Jobs to Cores per Node on Terra

A.

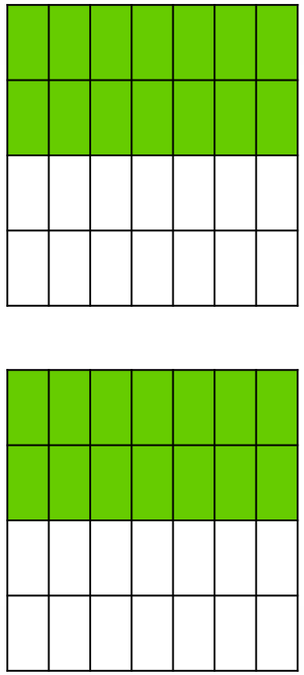


28 cores on
1 compute node

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

Preferred Mapping
(if applicable)

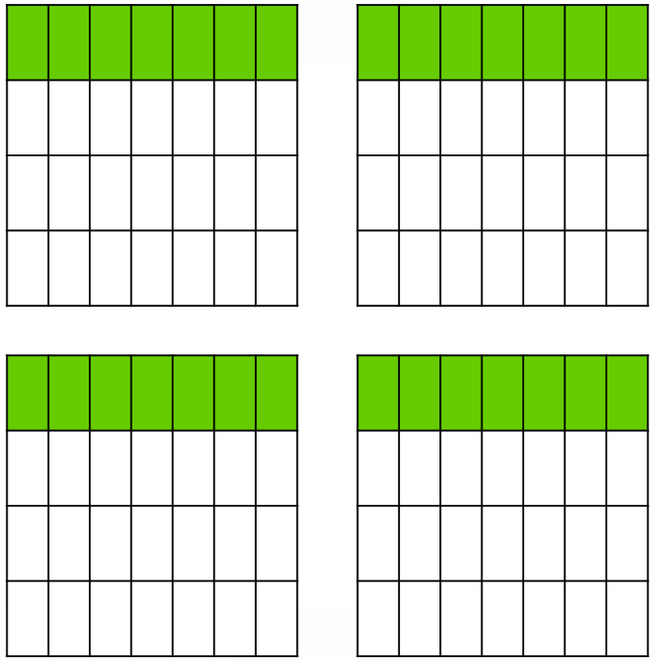
B.



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

Important Batch Job Parameters

Terra	Comment
`#SBATCH --export=NONE` `#SBATCH --get-user-env=L`	Initialize job environment.
`#SBATCH --time=HH:MM:SS`	Specifies the time limit for the job. Must specify seconds SS on Terra
`#SBATCH --ntasks=NNN`	Total number of tasks (cores) for the job.
`#SBATCH --ntasks-per-node=XX`	Specifies the maximum number of tasks (cores) to allocate per node

`#SBATCH --mem=nnnnM`
`or`
`#SBATCH --mem=nG`
 	
`(memory per NODE)`
	Sets the maximum amount of memory (MB).
	
	G for GB is supported on Terra

hprc.tamu.edu/wiki/HPRC:Batch_Translation

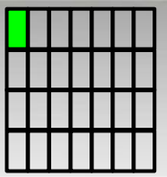
Pop Quiz

```
#SBATCH --export=NONE
#SBATCH --get-user-env=L
#SBATCH --job-name=stacks_S2
#SBATCH --ntasks 80
#SBATCH --ntasks-per-node=20
#SBATCH --mem=40G
#SBATCH --time=48:00:00
#SBATCH --output=/scratch/user/dylan/stdout.%J
#SBATCH --error stderr.%J
```

How many nodes is this job requesting?

A. 1600
B. 80
C. 20
D. 4

Terra Job File (Serial Example)



```
#!/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=JobExample1       # Set the job name to "JobExample1"
#SBATCH --time=01:30:00              # Set the wall clock limit to 1hr and 30min
#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]"

##OPTIONAL JOB SPECIFICATIONS
#SBATCH --account=123456             # Set billing account to 123456
#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
./myprogram
```

SUs = 1.5

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
```

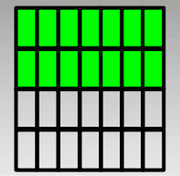
```
JOBID   NAME      USER      PARTITION  NODES  CPUS  STATE    TIME   TIME_LEFT  START_TIME          REASON      NODELIST
64039   somejob   someuser  medium     4      112   PENDING  0:00   20:00      2017-01-30T21:00:4  Resources
64038   somejob   someuser  medium     4      112   RUNNING  2:49  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

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

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

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

CRLF Line Terminators

Windows editors such as Notepad will add hidden Carriage Return Line Feed (CRLF) characters that will cause problems with many applications

```
cd $SCRATCH/batch_examples
file dos_text.txt
```

use `file` command to check

```
dos_text.txt: ASCII English text, with CRLF line terminators
```

```
cat -v dos_text.txt
```

use `cat` command to see CRLF characters

```
dos2unix dos_text.txt
file dos_text.txt
```

use `dos2unix` command to correct

```
dos_text.txt: ASCII English text
```

Need Help?

- First check the FAQ hprc.tamu.edu/wiki/HPRC:CommonProblems
  - Terra User Guide hprc.tamu.edu/wiki/Terra
    - Exercises hprc.tamu.edu/wiki/Terra:Exercises
  - Email your questions to **help@hprc.tamu.edu**. (Managed by a ticketing system)

- Help us, help you -- we need more info
  - Which Cluster
  - UserID/NetID (*UIN is not needed!*)
  - Job id(s) if any
  - Location of your jobfile, input/output files
  - Application used if any
  - Module(s) loaded if any
  - Error messages
  - Steps you have taken, so we can reproduce the problem

HIGH PERFORMANCE RESEARCH COMPUTING

TEXAS A&M UNIVERSITY

Thank you.

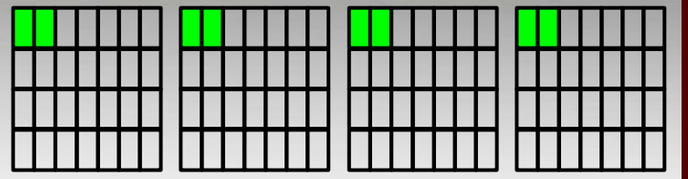
Any questions?

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/Terra:Batch#Environment_Variables

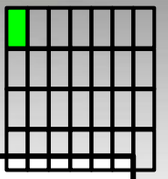
Terra Job File (multi core, multi node)



```
#!/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

File Transfers and **Terra**

- Simple File Transfers:

  Two-factor authentication required

  - scp: command line (Linux, Mac, Windows cmd)
  - rsync: command line (Linux, Mac, Windows); ***can resume transfer***
  - MobaXterm: GUI (Windows)
  - WinSCP: GUI (Windows)
  - Cyberduck: GUI (Mac)
  - Portal: portal.hprc.tamu.edu (web page; through "Files" menu)
  - rclone: move files to/from cloud storage; command line (HPRC clusters)
- Bulk data transfers:
  - Use fast transfer nodes
    - data transfer processes will not timeout at 60 minutes
    - on **Terra**: `terra-ftn.hprc.tamu.edu`
    - Globus Connect (hprc.tamu.edu/wiki/SW:GlobusConnect)

hprc.tamu.edu/wiki/HPRC:FileTransfers

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.