HIGH PERFORMANCE RESEARCH COMPUTING

HPRC Primers

Introduction to FASTER: A Composable Cluster

High Performance
Research Computing
DIVISION OF RESEARCH
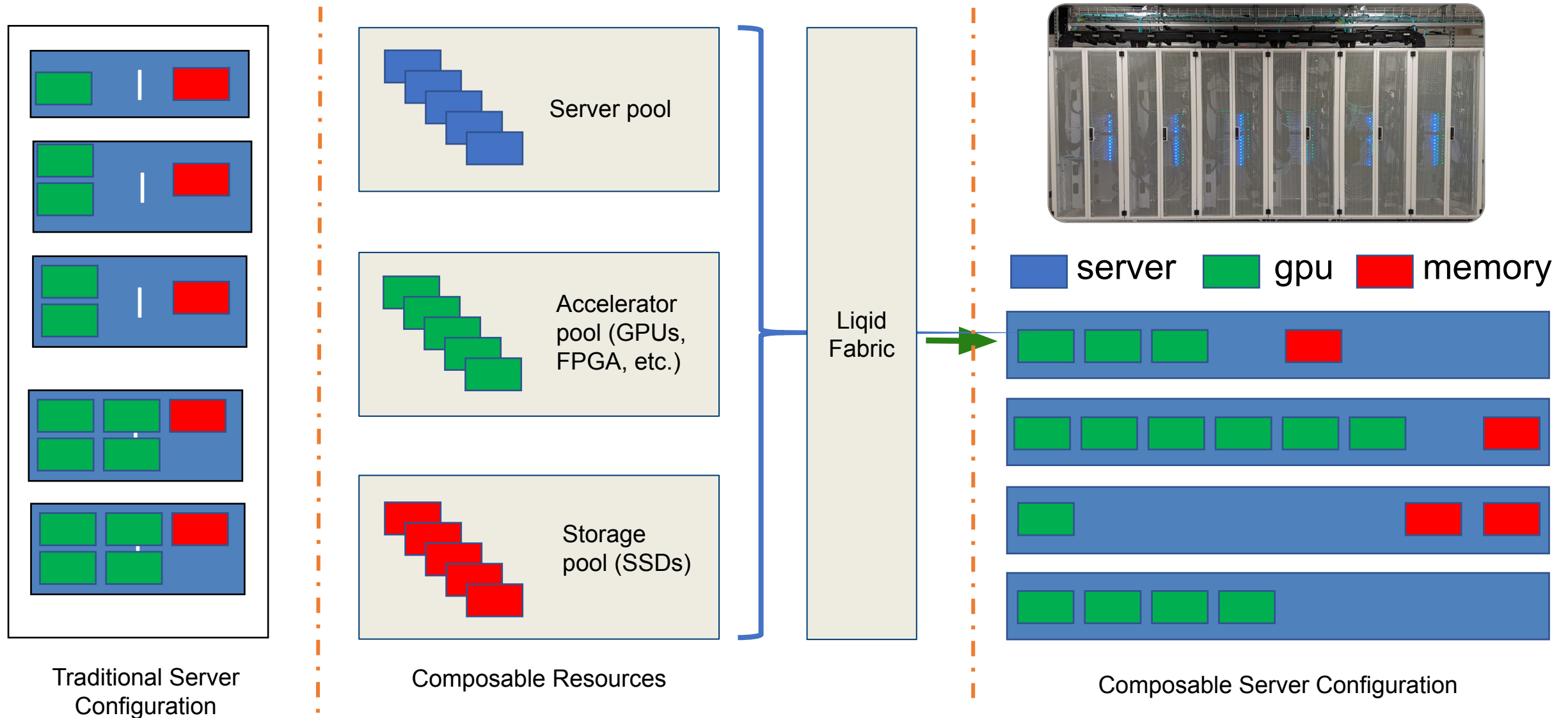
FASTER System Description

[FASTER](#) (Fostering Accelerated Sciences Transformation Education and Research) is a 180-node Intel cluster from Dell with an InfiniBand HDR-100 interconnect and Liqid PCIe Gen4 for composing the GPUs. NVIDIA A100, A10, A30, A40 and T4 GPUs are available. The 180 compute nodes are based on the Intel Ice Lake processor.

Node Type			
	Login	**Compute**	**Large Memory**
Quantity	3 (2 for TAMU, 1 for ACCESS)	180	2
CPU	2 32-core (64 cores) Intel Xeon 8352Y	2 32-core (64 cores) Intel Xeon 8352Y	2 32-core (64 cores) Intel Xeon 8352Y
Memory	256 GB	256 GB	1 TB
Additional Cluster Specifications			
Disk Space	3.84 TB NVMe (local, /tmp) 5PB DDN Lustre (global)		
Composable Accelerators	**GPU**: 200 T4 16GB; 40 A100 40GB; 10 A10 24GB; 4 A30 24GB; 8 A40 48GB		
Interconnect	Mellanox HDR100 InfiniBand (MPI and storage) Liqid PCIe Gen4 (resource composability)		

https://hprc.tamu.edu/wiki/FASTER:Intro

Composability at the Hardware Level

Traditional Server Configuration

Server pool

Accelerator pool (GPUs, FPGA, etc.)

Storage pool (SSDs)

Composable Resources

Liqid Fabric

server | gpu | memory

Composable Server Configuration

hprc.tamu.edu/wiki/FASTER:Intro

Resource Composability on FASTER

- GPUs can be added to compute nodes on the fly by using the "gres" option in a SLURM script.
- A researcher can request up to 12* GPUs to create these CPU-GPU nodes.
- This approach allows for more flexibility and options for node types (CPU only, CPU + GPU) available to users.
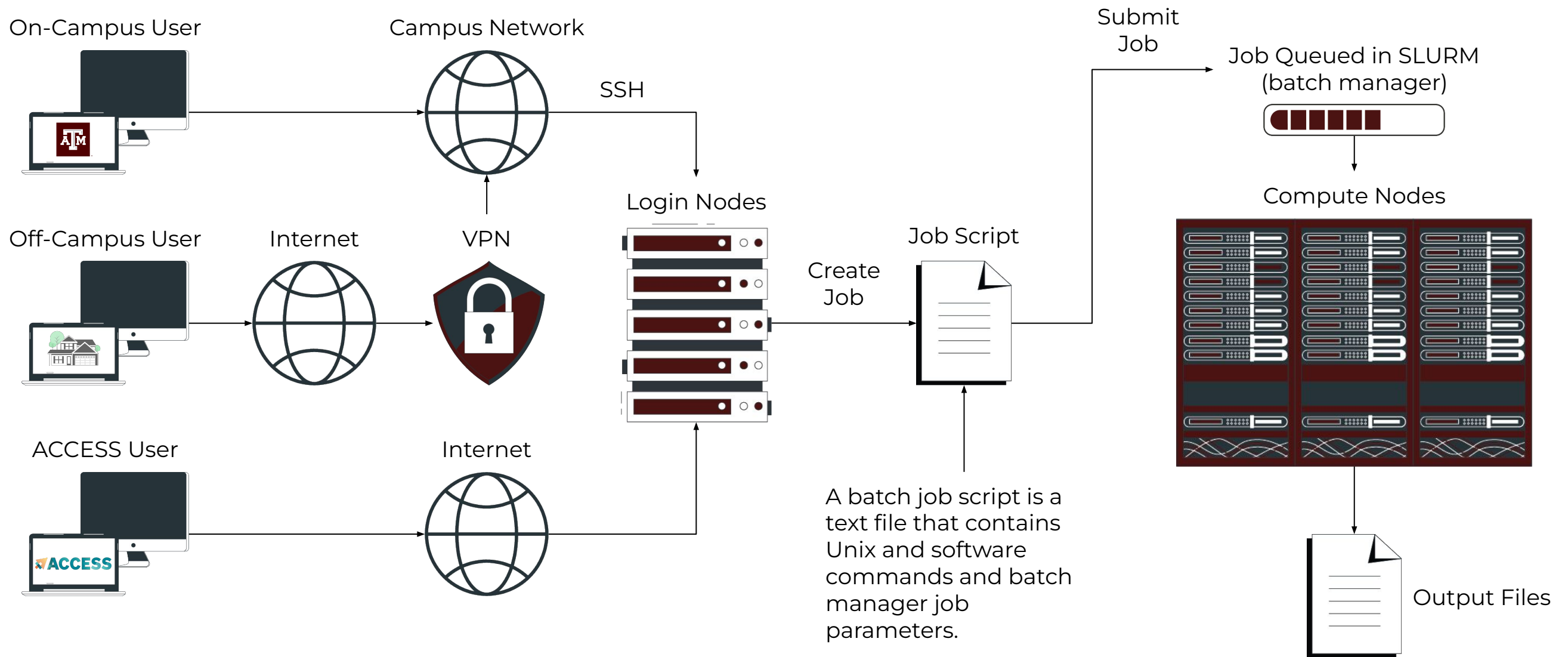
* Subject to change

Pop Quiz

Which accelerators are available on FASTER?

A. T4	B. A100	C. A10
D. A40	E. A and B	F. All the above

Batch Computing on HPRC Clusters

On-Campus User

Campus Network

Submit Job

Job Queued in SLURM (batch manager)

SSH

Compute Nodes

Login Nodes

Off-Campus User

Internet

VPN

Create Job

Job Script

ACCESS User

Internet

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

Output Files

Accessing FASTER via SSH

- SSH (Secure SHell) allows users to establish a connection between their local machine and the TAMU HPRC clusters.
- SSH Programs

Operating System	Windows	MacOS	Linux
Programs	MobaXTerm* PuTTY SSH Windows Subsystem for Linux (WSL) Windows Command Prompt	Terminal*	Terminal*
	* Recommended		

- SSH Command
  - TAMU (on-campus: Duo TFA; off-campus: VPN and Duo TFA)

```
ssh [NetID]@faster.hprc.tamu.edu
```

  - ACCESS (additional steps required)

```
ssh -J [username]@faster-jump.hprc.tamu.edu:8822 [username]@login.faster.hprc.tamu.edu
```

- https://hprc.tamu.edu/wiki/HPRC:Access

Two-Factor Authentication (TAMU)

- Enrollment in Duo Two-Factor Authentication is required for access to all TAMU HPRC computing resources: https://it.tamu.edu/duo/
  - Authenticated through CAS (Central Authentication System)
  - Additional information: https://hprc.tamu.edu/wiki/Two_Factor
- SSH clients tested with Duo by TAMU HPRC

Operating System	Windows	MacOS	Linux
Programs	MobaXTerm* PuTTY SSH Windows Subsystem for Linux (WSL) Windows Command Prompt	Terminal*	Terminal*
* Recommended			

https://hprc.tamu.edu/wiki/HPRC:Access

Accessing the HPRC Portal

- HPRC webpage: hprc.tamu.edu
  - TAMU: portal-faster.hprc.tamu.edu
  - ACCESS: portal-faster-access.hprc.tamu.edu

Accessing FASTER via the HPRC Portal (ACCESS)

Navigate to portal-faster-access.hprc.tamu.edu to get to the ACCESS CILogon OpenID Connect page.

Log-in using your ACCESS credentials. Create an account if you do not already have one.

Select the Identity Provider appropriate for your account.

Accessing FASTER via the HPRC Portal (ACCESS)

Complete the Duo Two-Factor Authentication prompt. Set-up the two-factor authentication if you have not already done so.
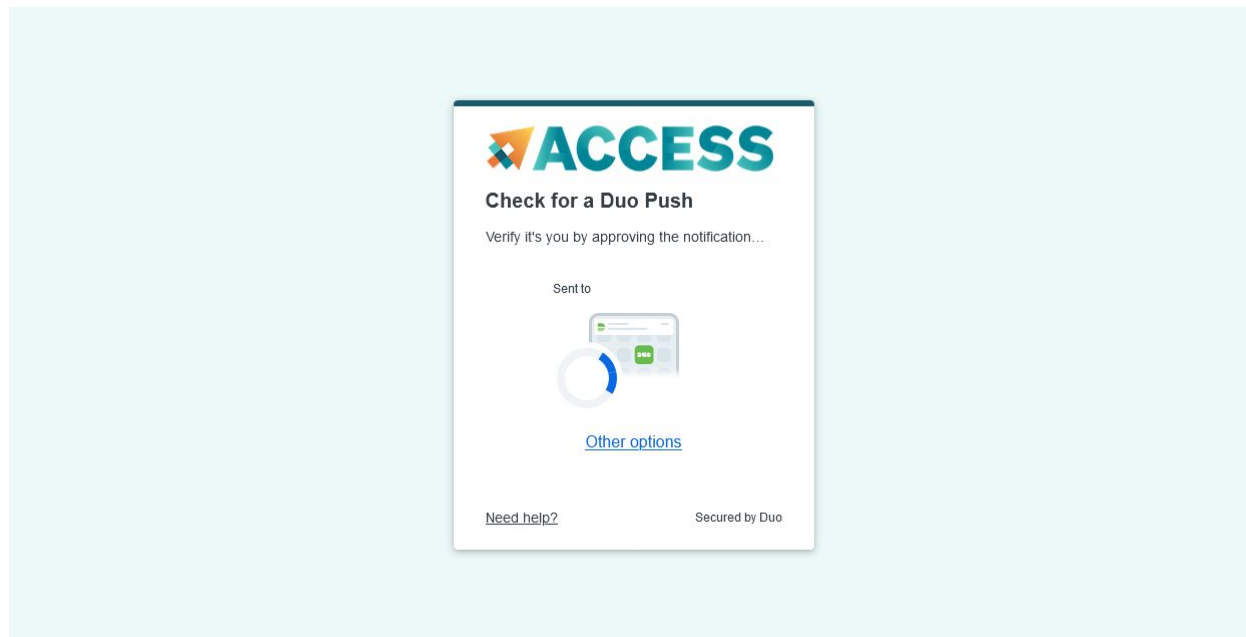
You will now have access to the FASTER portal.
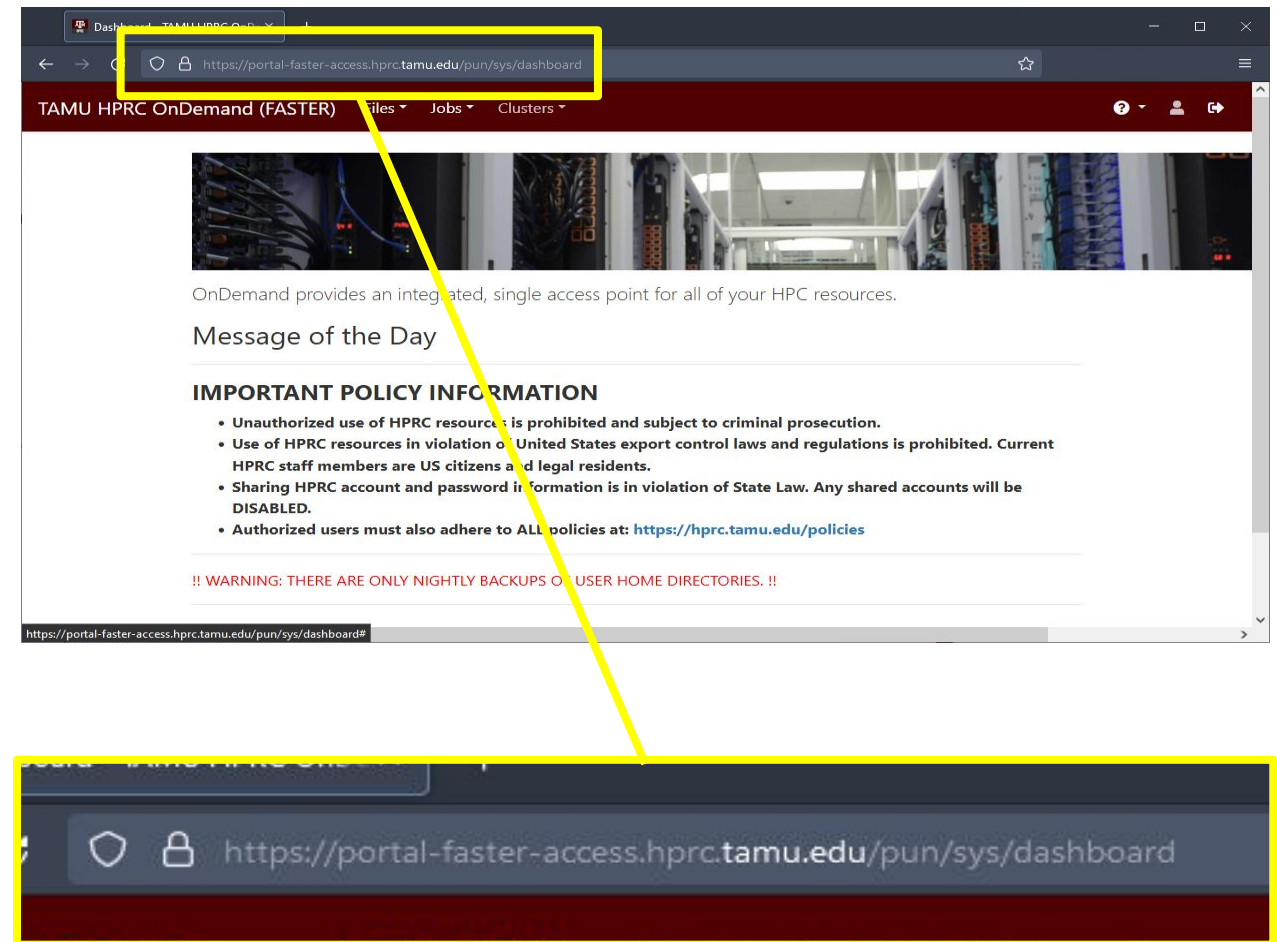
Hands-On Activity - 2 Minutes

1. Please try to access FASTER now through the portal.

2. What message do you see when you log on to the shell?

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.

$SCRATCH is shared between FASTER and Grace clusters.

View file usage and quota limits using the command: `showquota`

Do NOT share your home or scratch directories. Request a group directory for sharing files.

hprc.tamu.edu/wiki/FASTER:Filesystems_and_Files

Software

- See the Software wiki page for instructions and examples
  - https://hprc.tamu.edu/wiki/SW
  - https://hprc.tamu.edu/software/faster/
- License-restricted software
  - Contact help@hprc.tamu.edu
- Contact HPRC for software installation help/request
  - User can install software in their home/scratch directory
  - Do NOT run the sudo command when installing software

Software: Application Modules

Installed applications are made available with the module system
FASTER uses a software hierarchy inside the module system
In this hierarchy, the user loads a compiler toolchain which then
makes available software built with the currently loaded compiler
toolchain

```
module list                          # shows which software is available

module load GCCcore/12.1.0           # load GCC compiler version 12.1.0

module list                          # show which software is now available

module load OpenMPI/4.1.1            # load OpenMPI version 4.1.1
```

Software: Modules and Toolchains

- Toolchains are what we call groups of compilers & libraries
- There's a variety of toolchains on the clusters:

```
intel/2020b
iomkl/2021b
foss/2022a
GCCcore/12.1.0
```

- Module management:

```
module spider        # Search for modules and its dependencies
module purge         # removes all loaded modules from environment
```

Hands-On Activity - 2 Minutes

1. Please search for and load the following module:

```
GCC/12.1.0
```

2. Next remove all your current modules.

Consumable Computing Resources

- Resources specified in a job file:
  - Processor cores
  - Memory
  - Wall time
  - GPU
- Service Unit (SU)
  - 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

```
license_status -a
```

Find available license for "Matlab":

```
license_status -s Matlab
```

```
License status for Matlab:
-----------------------------------------------------------------
|License Name                      |  # Issued|  # In Use|# Available|
-----------------------------------------------------------------
|Matlab                            |        50|         0|        50|
-----------------------------------------------------------------
```

Find detail options:

```
license status -h
```

```
myproject
```

```
============================================================
            List of YourNetID's Project Accounts
------------------------------------------------------------
| Account      | FY  | Default | Allocation |Used & Pending SUs|  Balance  |    PI    |
------------------------------------------------------------
|1228000223136|  2023 |       N|    10000.00|              0.00|   10000.00|Doe, John |
|1428000243716|  2023 |       Y|     5000.00|            -71.06|    4928.94|Doe, Jane |
------------------------------------------------------------
```

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|  2023  |        N|   10000.00|             0.00|  10000.00|Doe, John |
    --------------------------------------------------------------------------------
    |1428000243716|  2023  |        Y|    5000.00|           -71.06|   4928.94|Doe, Jane |
    --------------------------------------------------------------------------------
    |1258000247058|  2023  |        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

Hands-On Activity - 2 Minutes

1. Use myproject to check the SU balance of your accounts.

2. Check the license status of Matlab.

   How many licenses are in the # Issued column?

Batch Queues

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

Additional information:
https://hprc.tamu.edu/wiki/FASTER:Batch#Batch_Queues

sinfo : Current Queues on FASTER

```
[abhinand@faster1 abhinand]$ sinfo
PARTITION        AVAIL   TIMELIMIT     JOB_SIZE    NODES(A/I/O/T)    CPUS(A/I/O/T)
cpu*             up      7-00:00:00    1-32        82/3/27/112       4697/743/1728/7168
gpu              up      7-00:00:00    1-32        14/27/11/52       474/2150/704/3328
memverge         up      2-00:00:00    1-2         0/0/2/2           0/0/128/128
fpga             up      2-00:00:00    1-2         0/0/2/2           0/0/128/128
spr              up      2-00:00:00    1           0/1/0/1           0/112/0/112
atsp             up      2-00:00:00    1           0/8/2/10          0/512/128/640
staff            down    7-00:00:00    1-infinite  96/38/44/178      5171/3405/2816/11392
```

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

```
#!/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 purge
module load GCCcore/11.3.0
module load Python/3.10.4


python my_program.py
```

These parameters describe your job to the job scheduler.

Account number to be charged.

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.

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 username`

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]

Hands-On Activity - 5 Minutes

1. Create a submission file for a serial job.
   Write the following below the line #run your program

   echo 'Hello World' > ExOutput.txt

2. Submit this job file using sbatch

3. Check which queue the job is in

4. When your job completes, check the contents of the ExOutput.txt file

Need Help?

First check the FAQ hprc.tamu.edu/wiki/HPRC:CommonProblems
- FASTER User Guide hprc.tamu.edu/wiki/FASTER
- Email your questions to help@hprc.tamu.edu

Help us, help you -- we need more info
- Which Cluster
- Username
- 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

Thank you.

Any questions?

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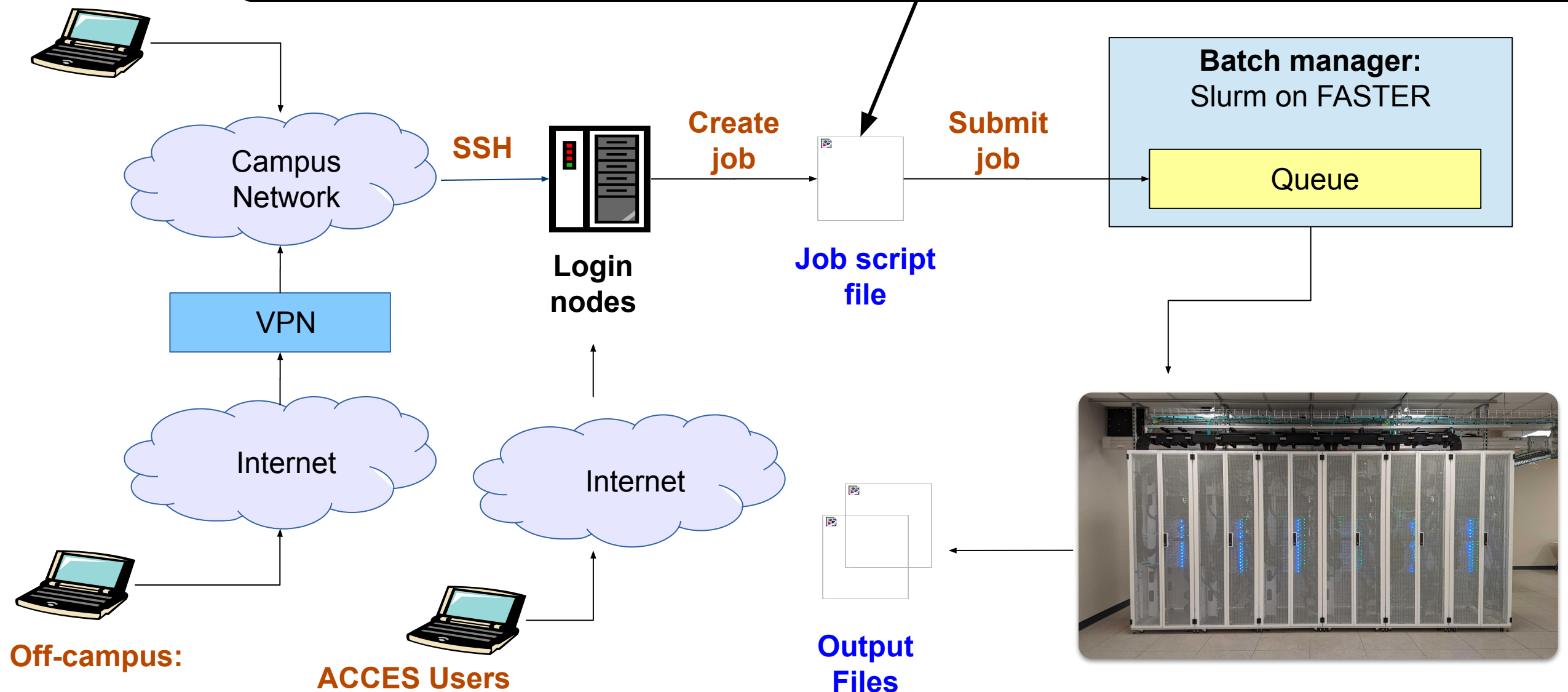
```
dos_text.txt: ASCII English text
```

Composability on FASTER

- GPUs can be added to compute nodes on the fly by using the "gres" option in a Slurm script.
- A researcher can request up to 12 GPUs to create these CPU-GPU nodes.
- This approach allows for more flexibility and options for node types (CPU only, CPU + GPU) available to users.

Batch Computing on HPRC Clusters

On-campus:

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

SSH

Login nodes

Create job

Job script file

Submit job

Batch manager:
Slurm on FASTER

Queue

Campus Network

VPN

Internet

Internet

Off-campus:

ACCES Users

Output Files

Cluster compute nodes

Accessing FASTER: via the Portal

- Access through (most) web browsers:
  portal.hprc.tamu.edu
    Top Banner Menu "Clusters" -> "FASTER Shell Access"
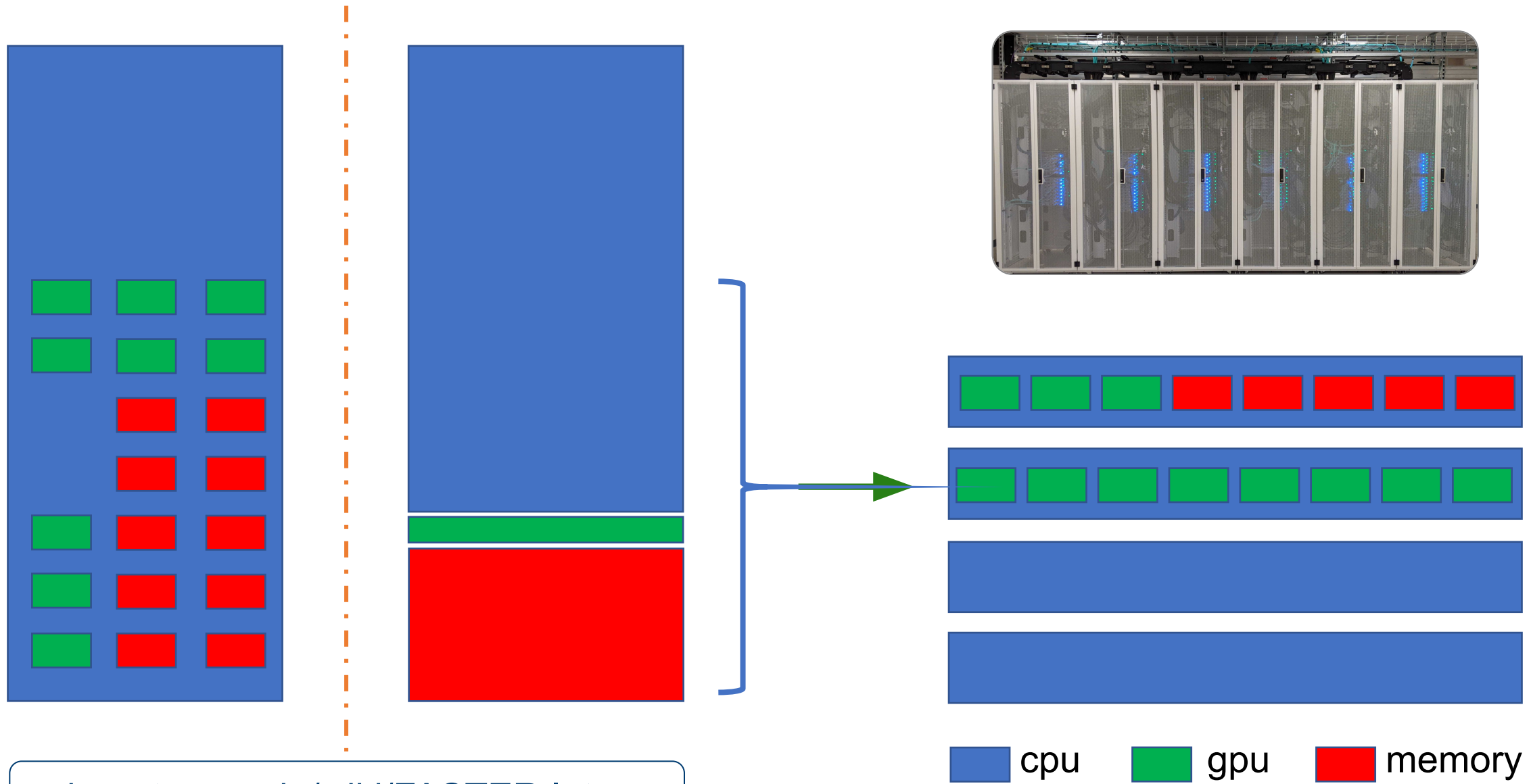

hprc.tamu.edu/wiki/HPRC:Access

FASTER

FASTER (Fostering Accelerated Sciences Transformation Education and Research) is a 180-node Intel cluster from Dell with an InfiniBand HDR-100 interconnect and Liqid PCIe Gen4 for composing the GPUs. NVIDIA A100, A10, A30, A40 and T4 GPUs are available. The 180 compute nodes are based on the Intel Ice Lake processor.

Node Type			
	Login	**Compute**	**Large Memory**
Quantity	3 (2 for TAMU, 1 for ACCESS)	180	2
CPU	2x 32-core (64 cores) Intel Xeon 8352Y	2x 32-core (64 cores) Intel Xeon 8352Y	2x 32-core (64 cores) Intel Xeon 8352Y
Memory	256 GB	256 GB	1 TB
Additional Cluster Specifications			
Disk Space	3.84 TB NVMe (local, /tmp); 5PB DDN Lustre (global)		
Composable Accelerators	**GPU**: 200 T4 16GB; 40 A100 40GB; 10 A10 24GB; 4 A30 24GB; 8 A40 48GB **FPGA**: 2 Intel D5005 PAC		
Interconnect	Mellanox HDR100 InfiniBand (MPI and storage) Liqid PCIe Gen4 (GPU composability)		

For more information:
https://hprc.tamu.edu/wiki/FASTER:Intro

Composability at the Hardware Level

cpu   gpu   memory

Composability at the Hardware Level

server gpu memory

High Performance Research Computing | hprc.tamu.edu | NSF Award #2112356
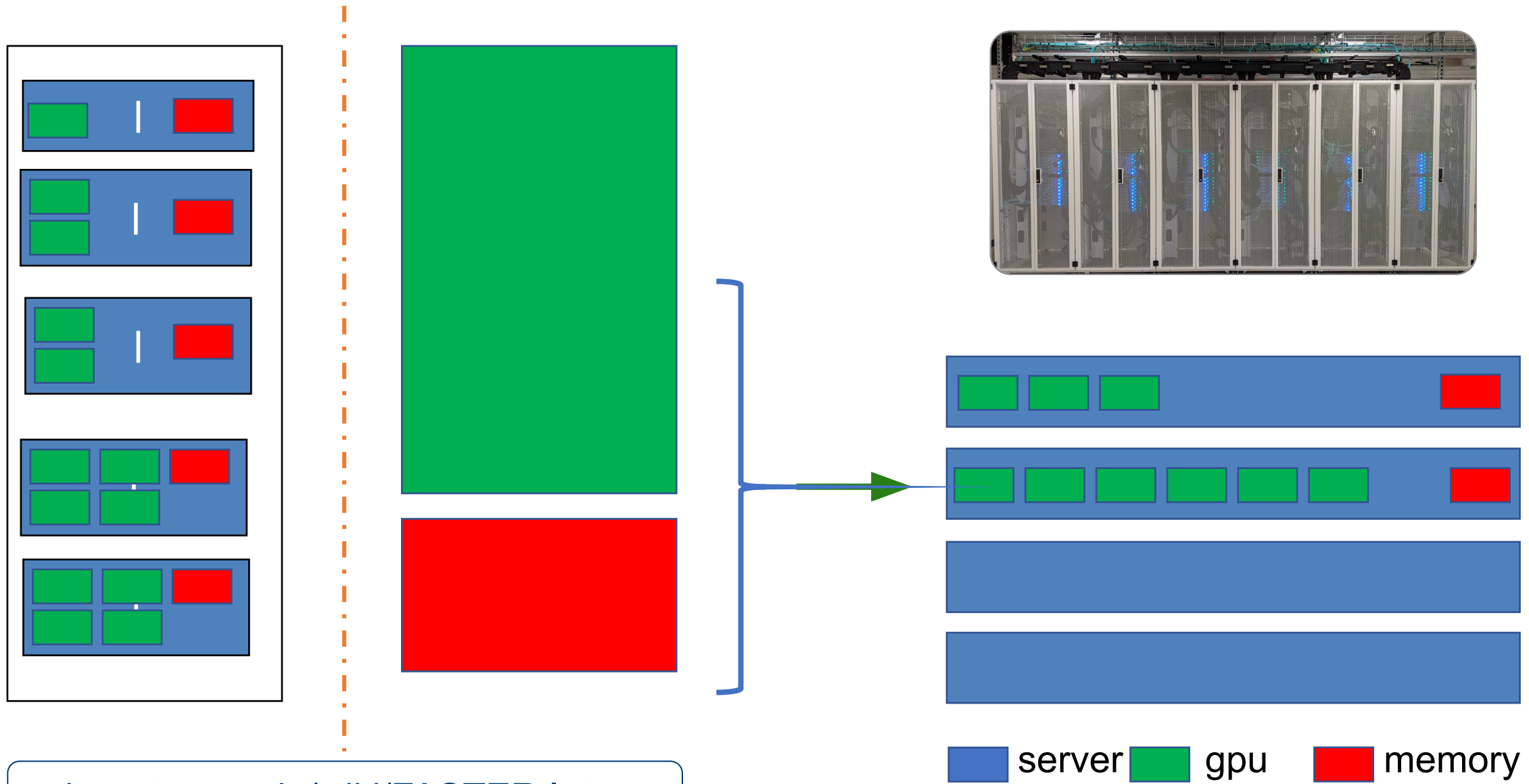
Accessing FASTER via SSH (TAMU)

- SSH command is required for accessing FASTER:
  - On campus: `ssh userNetID@faster.hprc.tamu.edu`
  - Off campus:
    - Set up and start VPN (Virtual Private Network):   u.tamu.edu/VPnetwork
    - Then: `ssh userNetID@faster.hprc.tamu.edu`
  - *Two-Factor Authentication* enabled for CAS, VPN, SSH

- SSH programs for Windows:
  - MobaXTerm (preferred, includes SSH and X11)
  - PuTTY SSH
  - Windows Subsystem for Linux
- FASTER has 3 login nodes. Check the bash prompt.

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

Accessing FASTER via SSH

- SSH (Secure SHell) allows users to establish a connection between their local machine and the TAMU HPRC clusters.
- SSH Programs

Operating System	Windows	MacOS	Linux
Programs	MobaXTerm* PuTTY SSH Windows Subsystem for Linux (WSL)	Terminal*	Terminal*
* Recommended			

- SSH Command
  - TAMU (on-campus: Duo TFA; off-campus: VPN and Duo TFA)
    ```
    ssh userNetID@faster.hprc.tamu.edu
    ```
  - ACCESS (additional steps required)
    ```
    ssh -J [username]@faster-jump.hprc.tamu.edu:8822 [username]@login.faster.hprc.tamu.edu
    ```
- Additional information: https://hprc.tamu.edu/wiki/HPRC:Access

Two-Factor Authentication (TAMU)

- 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

- <u>Not all software supports SSH+Duo: SFTP in Matlab</u>

hprc.tamu.edu/wiki/Two_Factor

Example: SSH login with Duo
```
$ ssh userNetID@faster.hprc.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

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

Accessing FASTER via SSH (ACCESS)

Rough outline, fixing later (Shaina):

Generate pub key

Submit pubkey to keys@hprc.tamu.edu

    Make sure it is pub key not private key

Wait for approval email

Log in via jump host

Sample Job Script Structure (**FASTER**)

```bash
#!/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 purge
module load Python/3.7.0-intel-2018b


./my_program.py
```

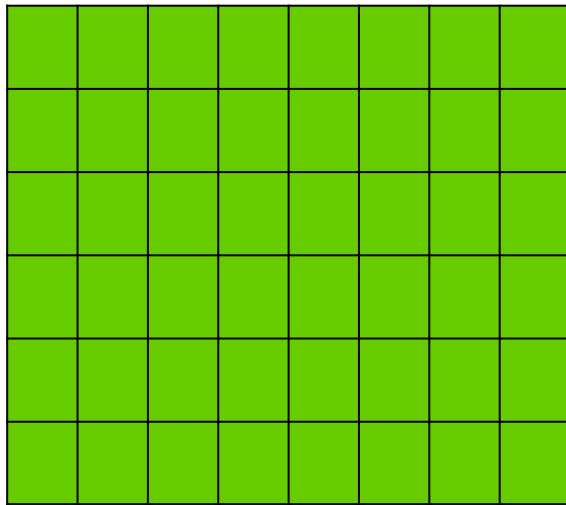These parameters describe your job to the job scheduler

Account number to be charged

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
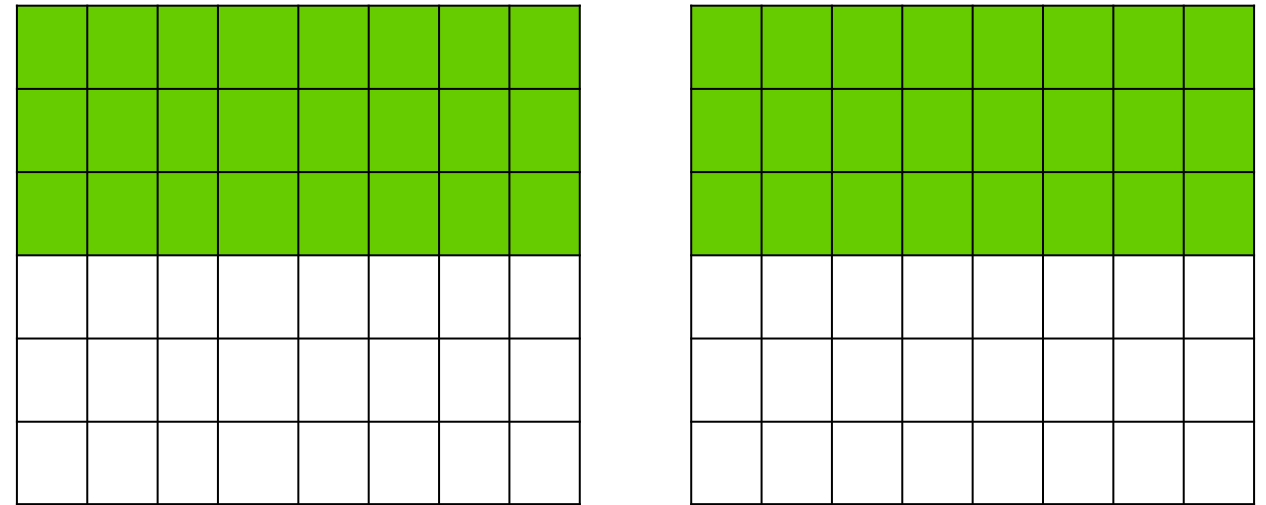
Mapping Jobs to Cores per Node on **FASTER**

A.

64 cores on
1 compute node

#SBATCH --ntasks=64
#SBATCH --tasks-per-node=64

Preferred Mapping
(if applicable)

B.

64 cores on
2 compute nodes

#SBATCH --ntasks=64
#SBATCH --tasks-per-node=32

Important Batch Job Parameters

FASTER	Comment
`#SBATCH --export=NONE` `#SBATCH --get-user-env=L`	Initialize job environment.
`#SBATCH --time=HH:MM:SS`	Specifies the time limit for the job.
`#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 FASTER

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   C. 20

B. 80   D. 4

FASTER Job File (Serial Example)

```bash
#!/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 purge
module load intel/2018b


# run your program
./myprogram
```

SUs = 1.5

Job Memory Requests on FASTER

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

Job Submission and Tracking

FASTER command	Description
sbatch jobfile	Submit jobfile 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

Job submission issue: insufficient SUs

```
$ 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
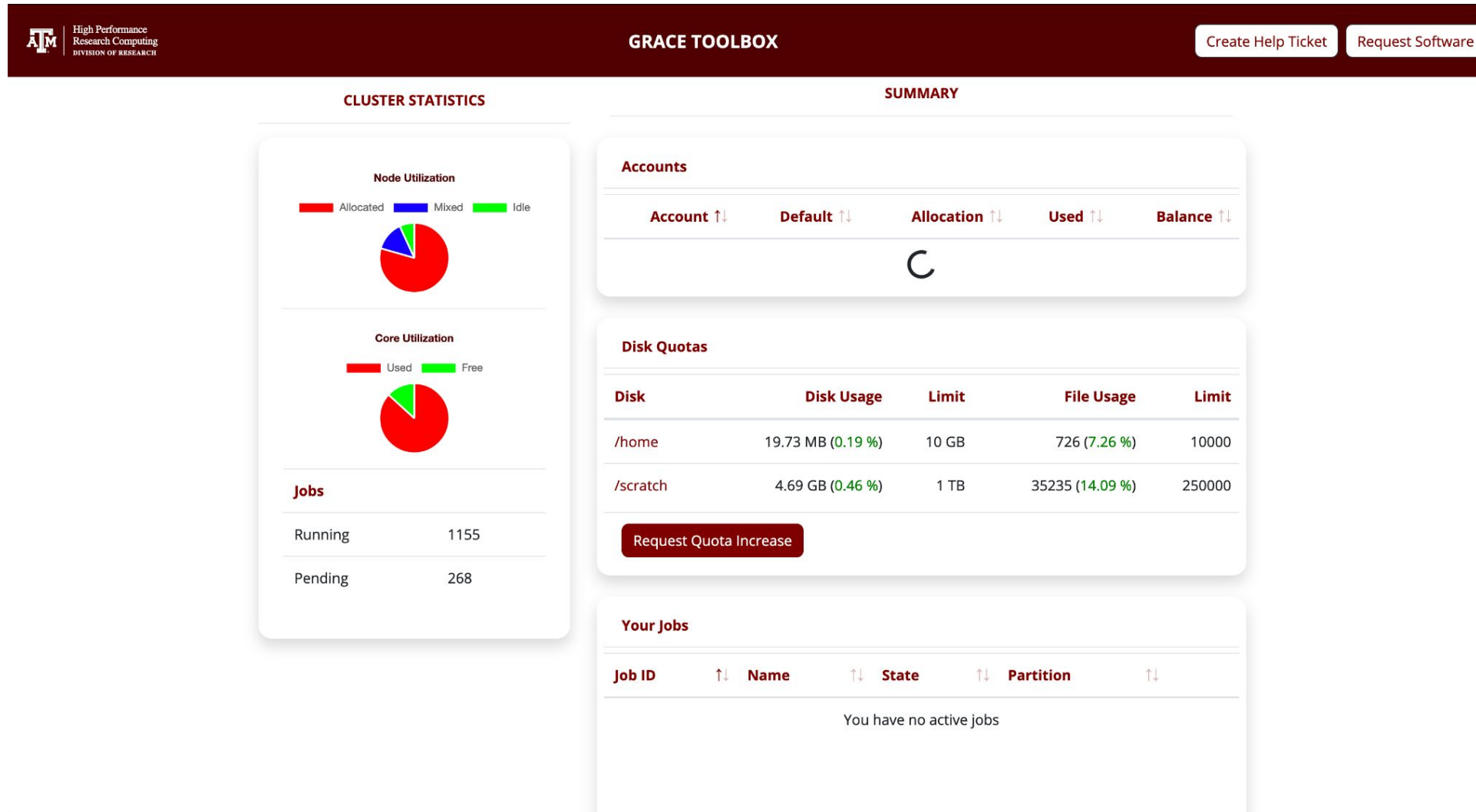- Apply for more SUs (if you are eligible, as a PI or permanent researcher)

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

OOD Dashboard: FASTER

Easily view Cluster utilization, Storage Quotas & Allocation Balances

Quota and file limit increases will only be considered for scratch directories

Preferred way to request *Quota Increases*