TEXAS TECH UNIVERSITY
Information Technology Division™

Introduction to the
Texas Tech RedRaider Cluster

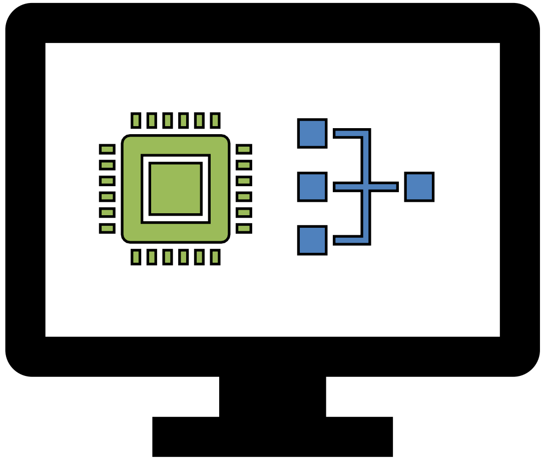
Misha Ahmadian
High Performance Computing Center
(on behalf of the HPCC staff)

Jan 19th, 2021

Outline

- ❖ Introducing the New HPCC Resources
- ❖ HPCC Software Environment
- ❖ Logging and using the RedRaider Cluster
- ❖ Resource Allocation and Job Submission with SLURM
- ❖ Software builds and installation
- ❖ Getting Help

Introducing the New HPCC Resources



HPCC Resources: Historical

TEXAS TECH UNIVERSITY
Information Technology Division™

Previous Clusters:

- Hrothgar
  - Commissioned in 2011
  - Decommissioned in Nov 2019
- Ivy
  - Commissioned in 2014
  - 100 nodes
  - 2000 total Cores (20 cores/node)
  - 6.25 TB Total RAM (64 GB/node)
  - Xeon E5-2670v2 **Ivy** Bridge Processors
  - QDR 40 GB/second InfiniBand fabric

HPCC Resources: Current

TEXAS TECH UNIVERSITY
Information Technology Division™

Previous Clusters:

- Quanah
  - Commissioned in 2017
  - 467 nodes
  - 16,812 Cores (36 cores/node)
  - 87.56 TB Total RAM (192 GB/node)
  - Intel Xeon E5-2695v4 Broadwell Processors
  - Non-blocking Omni-Path (100 Gbps) Fabric
  - Benchmarked at 485 Teraflops

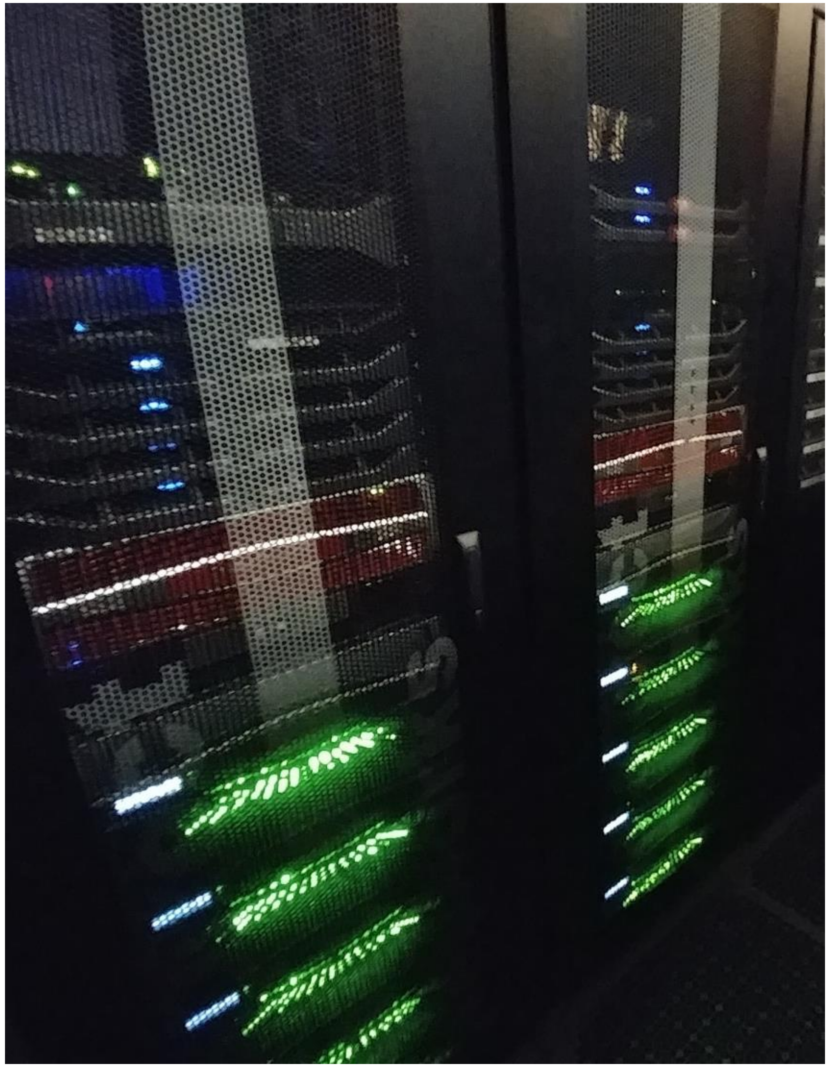
HPCC Resources: Storage

Lustre Storage System:

- Upgraded to 200 Gbps HDR Fabric
- 6.9 PB of storage space
- Quota/Backup/Purge per Lustre area:

Area	Quota	Backup	Purge
/home/<eraider>	300 GB	**Yes**	No
/lustre/work/<eraider>	700 GB	No	No
/lustre/scratch/<eraider>	None	No	**Monthly**

- User may purchase dedicated storage space
  - With Backup: $80/TB/Year
  - Without Backup $40/TB/Year



HPCC Resources: New

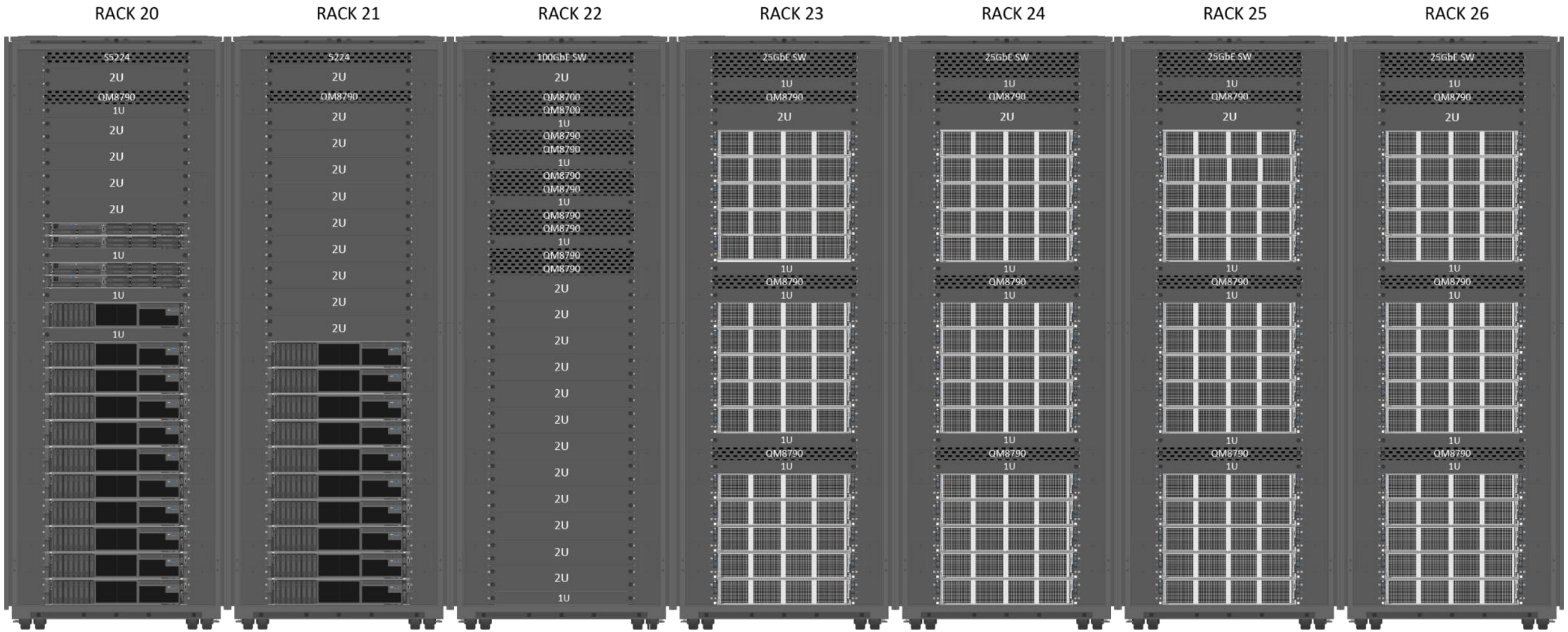
New Cluster Design Goal:

- More Compute Capacity
  - Add ~1 Petaflops total computing capacity beyond existing clusters
- Fit within existing limits
  - Accommodate to the existing cooling capacity
  - Fit within recently expanded power limits
- Coalesce the operation of existing clusters
  - Operate as a single cluster by combining the new cluster with the existing Quanah, Ivy, and Community Cluster nodes. (By January 2021)
- Connect all components to the central storage
  - Utilizing new LNet routers and expanding the storage network based on 200 Gbps Mellanox HDR fabric

HPCC Resources: New

TEXAS TECH UNIVERSITY
Information Technology Division™

- New RedRaider cluster:
  - Delivered by July 2020



HPCC Resources: New

New RedRaider Cluster Additions: Nocona CPU and Matador GPU

- Initial Installation

Front View

Back View

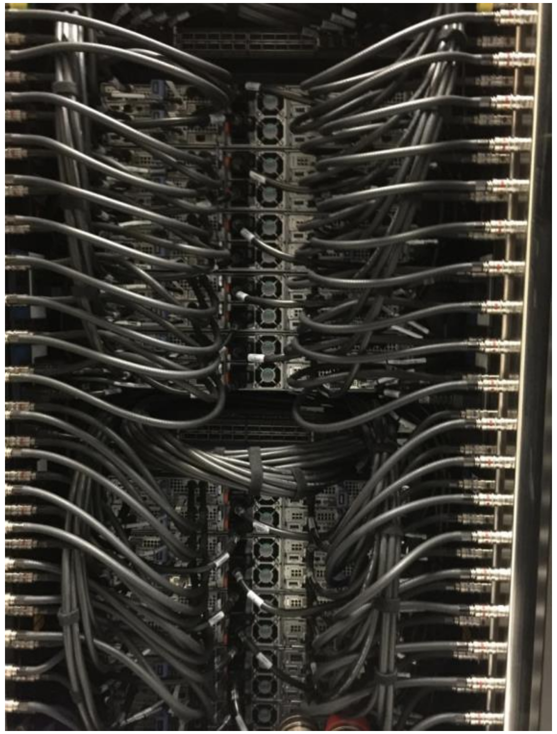
HPCC Resources: New

TEXAS TECH UNIVERSITY
Information Technology Division™

New RedRaider Cluster:

- Liquid Cooling installation for CPU nodes

Cooling Line Installation



Back view of cooling lines

Interior of CPU worker node

HPCC Resources: New

New (RedRaider) Cluster Components:

- **240 CPU nodes (Nocona)**
  - 30,720 Cores (128 cores/node)
  - 120 TB total RAM (512 GB/node)
  - AMD EPYC ROME 7702 processor
  - 804 Teraflops (81.4% efficiency)
- **20 GPU nodes (Matador)**
  - 40 NVIDIA Tesla V100 GPUs (2 V100 / node)
  - 7.5 TB total RAM (384 GB/node)
  - 800 CPU Cores (40 cores/node)
  - Intel Xeon Cascade lake 6248 processor
  - 226 Teraflops (80.6% Efficiency)
- **HDR 200 Gbps InfiniBand fabric**
- **Has been merged with Quanah cluster already**

HPCC Software Environment

HPCC Software Environment

	Ivy*	Quanah	RedRaider
Operating System	- CentOS 7.4	- CentOS 7.4 **	- CentOS 8.1
Job Resource Manager	- Slurm 20.11.0	- Slurm 20.11.0	- Slurm 20.11.0
Package Build Env	- RPM Build	- RPM Build	- Spack v0.15
Software Deployment Env	- Lmod 7.7.14	- Lmod 7.7.14	- Lmod 8.2.10
Available C/C++/Fortran /MPI Compilers	- GCC 4.8.5 (Default) - GCC 5.4.0 - GCC 7.3.0 - Intel 18.0.3.222 - impi 2018.3.222 - OpenMPI 1.10.[6-7] - MVAPICH 2.2	- GCC 4.8.5 (Default) - GCC 5.4.0 - GCC 7.3.0 - Intel 18.0.3.222 - impi 2018.3.222 - OpenMPI 1.10.[6-7] - MVAPICH 2.2	- GCC 8.3.1 (Default) - GCC 9.2.0 - GCC 10.1.0 (Recommended) - AOCC/AOCL (Coming Soon) - Intel compiler for GPU nodes (Coming Soon) - OpenMPI 3.1.6, 4.0.4 - MVAPICH & impi (Coming Soon)
GPU Libraries	- N/A	- N/A	- CUDA 11.0 (default) - Cudnn 8.0.1 (default)

* To be devoted to Open OnDemand   ** Upgrade to CentOS 8 soon

HPCC Software Environment

Program	Version	Program	Version	Program	Version	Program	Version
GCC 10	10.1.0	Netcdf-C-MPI	4.7.3	gls	2.5	root	6.18.4
GCC 9	9.2.0	Netcdf-CXX-MPI	4.3.1	boost	1.74.0	geant4	10.6.2
OpenMPI-3	3.1.6	Netcdf-Fort-MPI	4.5.2	Bowtie2	2.3.5.1	fastx-toolkit	0.0.14
OpenMPI-4	4.0.4	OpenBlas	0.3.10	Lammps	20200505	VASP	5.4.4
Singularity	3.5.3	OpenBlas-MPI	0.3.10	rmblast	2.9.0		
Python3	3.8.3	Lapack	3.8.3	samtools	1.1		
Perl	5.30.3	ScalaPack	2.1.0	bcftools	1.10.2		
R	4.0.2	Hdf5	1.10.6	bedtools	2.27.1		
Matlab	R2020b	Hdf5-MPI	1.10.6	mafft	7.453		
Java	11.0.2	udunits	2.2.24	GROMACS	2020.2		
Netcdf-C	4.7.3	nco	4.7.9	emboss	6.6.0		
Netcdf-Fortran	4.5.2	fftw	3.3.8	gnuplot	5.2.8		
Parallel-Netcdf	1.12.1	fftw-MPI	3.3.8	bwa	0.7.17		

Logging and Using the RedRaider Cluster

Getting Started

- User Guides:
  - http://www.depts.ttu.edu/hpcc/userguides/index.php
- More details about HPCC equipment:
  - http://www.depts.ttu.edu/hpcc/operations/equipment.php
- Logging Into the HPCC Resources:
  - User Guide: http://tinyurl.com/ttu-hpcc-login
  - Are you on or off campus?
  - Logging in from off campus:
    - Log in via SSH gateway
    - Establish a VPN connection - https://goo.gl/4LbuWG

Logging to RedRaider Cluster

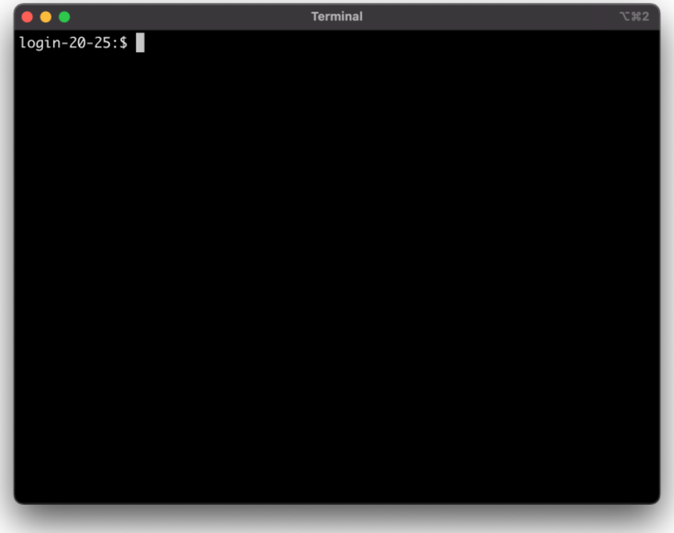
- Mac/Linux Users:
  - **SSH** (Secure Shell): Freely available on Linux/Unix/MacOS and used via the Terminal.

```
ssh eraider@login.hpcc.ttu.edu
```

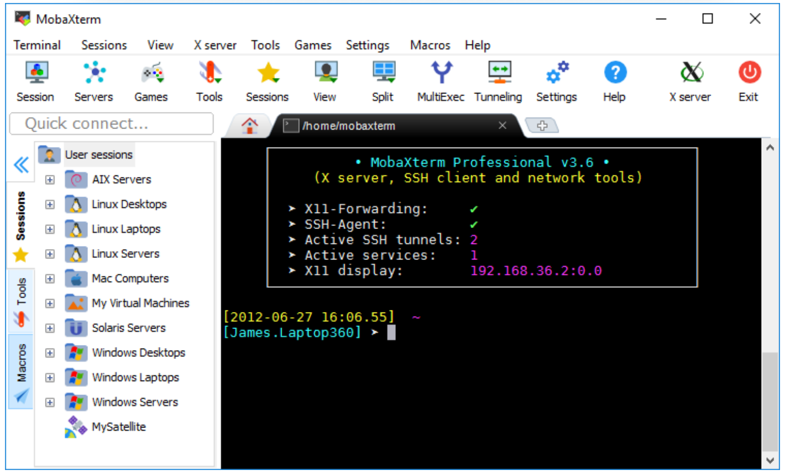
  - *The* ***`quanah.hpcc.ttu.edu`*** *login node is still available.*

- Windows Users:
  - **MobaXterm** (Recommended): https://mobaxterm.mobatek.net
  - **Putty**: https://www.putty.org

- After Logged in:
  - RedRaider has two login nodes: (**`login-20-25`**, **`login-20-26`**)
  - The load-balancer lands your SSH session on one of these nodes.



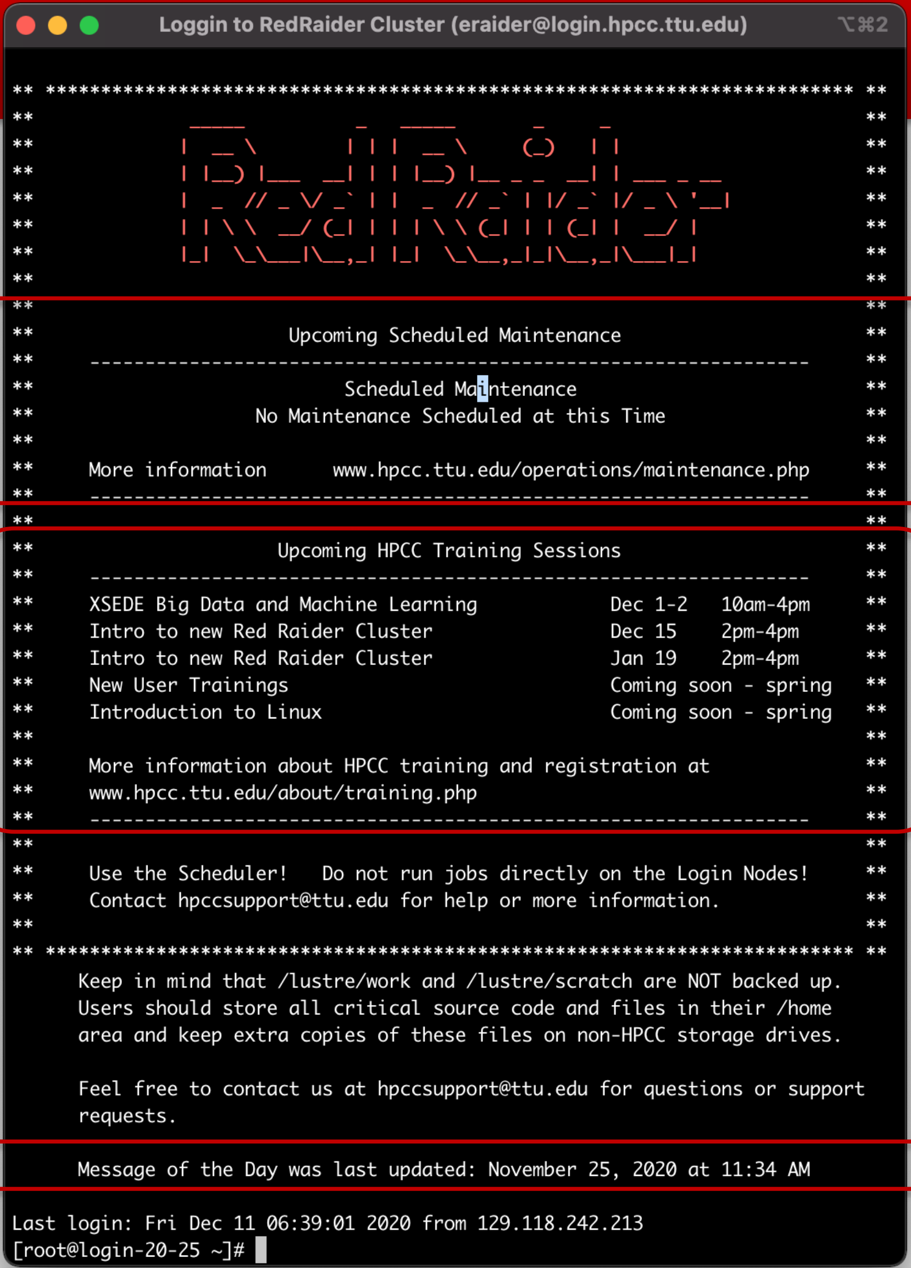
```
login-20-25:$
```



```
        • MobaXterm Professional v3.6 •
     (X server, SSH client and network tools)

   ➤ X11-Forwarding:      ✔
   ➤ SSH-Agent:           ✔
   ➤ Active SSH tunnels:  2
   ➤ Active services:     1
   ➤ X11 display:         192.168.36.2:0.0

[2012-06-27 16:06.55]  ~
[James.Laptop360] ➤
```

Logging In

TEXAS TECH UNIVERSITY
Information Technology Division™



```
** ************************************************************************* **
**                                                                           **
**              _____          _ _____       _     _                         **
**             |  __ \        | |  __ \     (_)   | |                        **
**             | |__) |___  __| | |__) |__ _ _  __| | ___ _ __               **
**             |  _  // _ \/ _` |  _  // _` | |/ _` |/ _ \ '__|              **
**             | | \ \  __/ (_| | | \ \ (_| | | (_| |  __/ |                 **
**             |_|  \_\___|\__,_|_|  \_\__,_|_|\__,_|\___|_|                 **
**                                                                           **
**                                                                           **
**                       Upcoming Scheduled Maintenance                      **
**     -----------------------------------------------------------------     **
**                            Scheduled Maintenance                          **
**                    No Maintenance Scheduled at this Time                  **
**                                                                           **
**     More information      www.hpcc.ttu.edu/operations/maintenance.php     **
**     -----------------------------------------------------------------     **
**                                                                           **
**                      Upcoming HPCC Training Sessions                      **
**     -----------------------------------------------------------------     **
**     XSEDE Big Data and Machine Learning            Dec 1-2   10am-4pm     **
**     Intro to new Red Raider Cluster                Dec 15    2pm-4pm      **
**     Intro to new Red Raider Cluster                Jan 19    2pm-4pm      **
**     New User Trainings                             Coming soon - spring   **
**     Introduction to Linux                          Coming soon - spring   **
**                                                                           **
**     More information about HPCC training and registration at              **
**     www.hpcc.ttu.edu/about/training.php                                   **
**     -----------------------------------------------------------------     **
**                                                                           **
**     Use the Scheduler!  Do not run jobs directly on the Login Nodes!      **
**     Contact hpccsupport@ttu.edu for help or more information.             **
**                                                                           **
** ************************************************************************* **
      Keep in mind that /lustre/work and /lustre/scratch are NOT backed up.
      Users should store all critical source code and files in their /home
      area and keep extra copies of these files on non-HPCC storage drives.

      Feel free to contact us at hpccsupport@ttu.edu for questions or support
      requests.

      Message of the Day was last updated: November 25, 2020 at 11:34 AM

Last login: Fri Dec 11 06:39:01 2020 from 129.118.242.213
[root@login-20-25 ~]#
```

Upcoming or current downtimes →

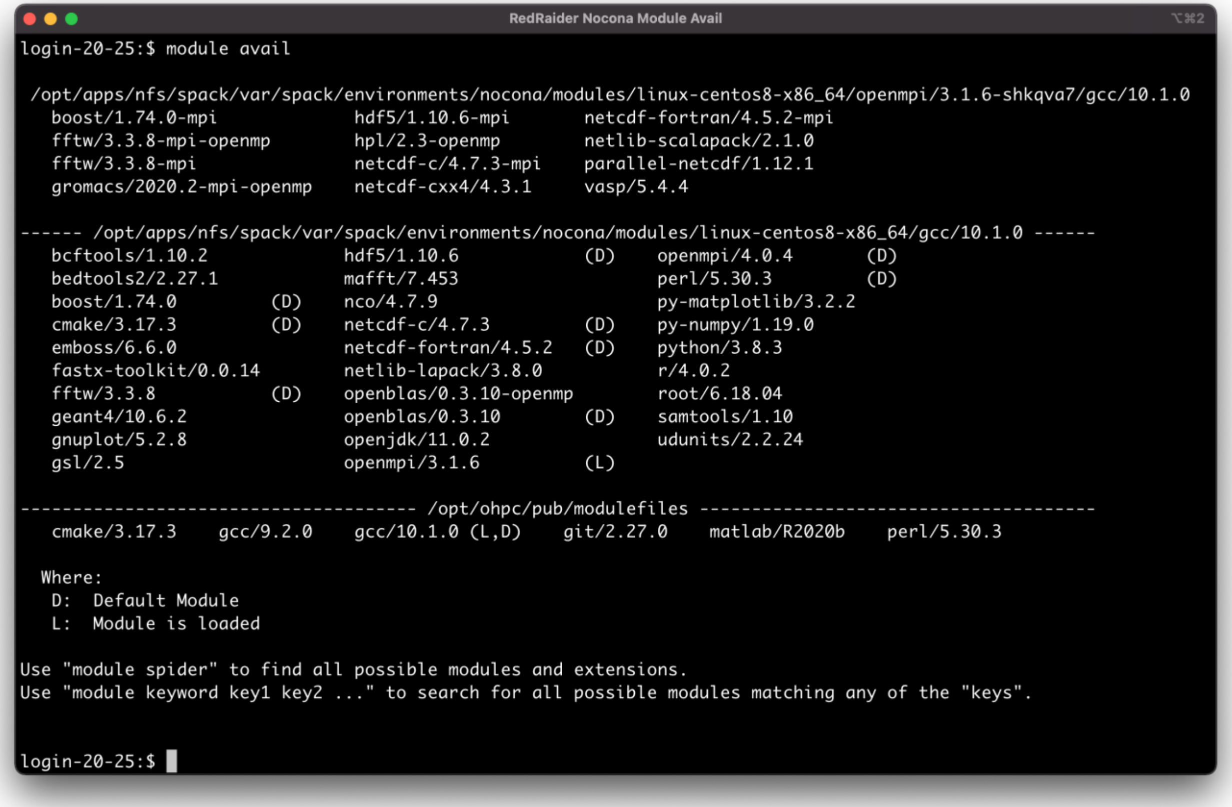
← Upcoming HPCC Training Sessions

Last updated time →

Environment Settings

TEXAS TECH UNIVERSITY
Information Technology Division™

- Lmod Modules:
  - The primary way to change your user environment
  - Please note that Quanah (Intel nodes), Nocona (AMD nodes) and Matador (GPU nodes) have different set of modules
  - Module commands:
    - `module avail`
    - `module list`
    - `module load <module_name>`
    - `module unload <module_name>`
    - `module spider <keyword>`
    - `module purge`



```
login-20-25:$ module avail

 /opt/apps/nfs/spack/var/spack/environments/nocona/modules/linux-centos8-x86_64/openmpi/3.1.6-shkqva7/gcc/10.1.0
   boost/1.74.0-mpi             hdf5/1.10.6-mpi       netcdf-fortran/4.5.2-mpi
   fftw/3.3.8-mpi-openmp        hpl/2.3-openmp        netlib-scalapack/2.1.0
   fftw/3.3.8-mpi               netcdf-c/4.7.3-mpi    parallel-netcdf/1.12.1
   gromacs/2020.2-mpi-openmp    netcdf-cxx4/4.3.1     vasp/5.4.4

------ /opt/apps/nfs/spack/var/spack/environments/nocona/modules/linux-centos8-x86_64/gcc/10.1.0 ------
   bcftools/1.10.2             hdf5/1.10.6            (D)    openmpi/4.0.4       (D)
   bedtools2/2.27.1            mafft/7.453                   perl/5.30.3         (D)
   boost/1.74.0         (D)    nco/4.7.9                     py-matplotlib/3.2.2
   cmake/3.17.3         (D)    netcdf-c/4.7.3         (D)    py-numpy/1.19.0
   emboss/6.6.0                netcdf-fortran/4.5.2   (D)    python/3.8.3
   fastx-toolkit/0.0.14        netlib-lapack/3.8.0           r/4.0.2
   fftw/3.3.8           (D)    openblas/0.3.10-openmp        root/6.18.04
   geant4/10.6.2               openblas/0.3.10        (D)    samtools/1.10
   gnuplot/5.2.8               openjdk/11.0.2                udunits/2.2.24
   gsl/2.5                     openmpi/3.1.6          (L)

-------------------------------------- /opt/ohpc/pub/modulefiles --------------------------------------
   cmake/3.17.3    gcc/9.2.0    gcc/10.1.0 (L,D)    git/2.27.0    matlab/R2020b    perl/5.30.3

  Where:
   D:  Default Module
   L:  Module is loaded

Use "module spider" to find all possible modules and extensions.
Use "module keyword key1 key2 ..." to search for all possible modules matching any of the "keys".


login-20-25:$ █
```

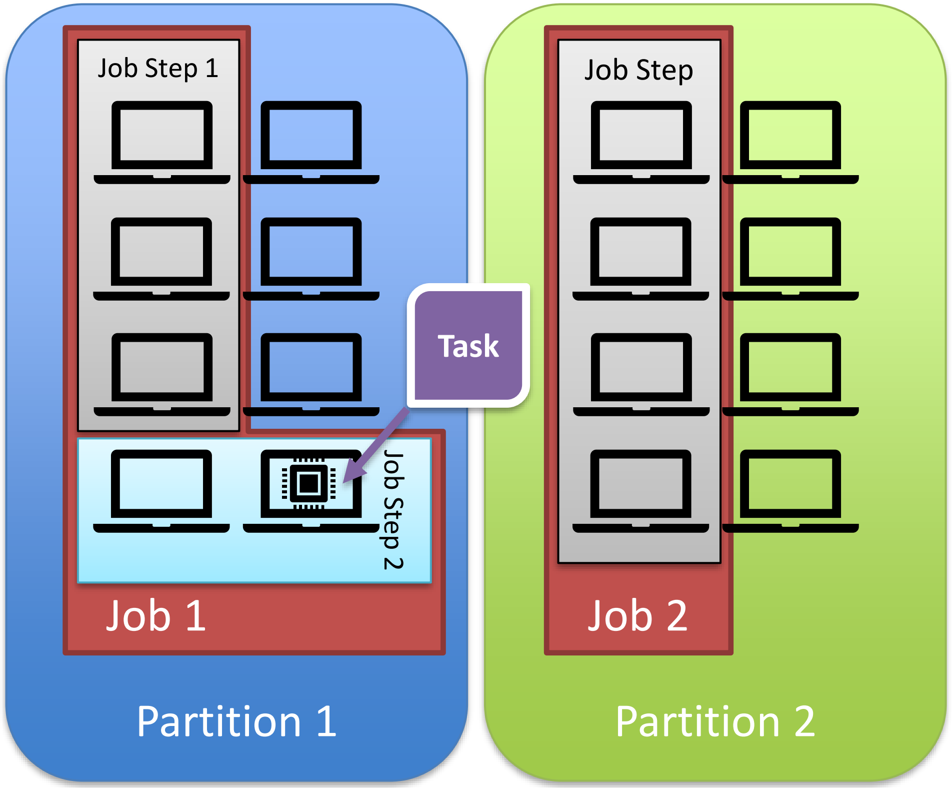
Resource Allocation and Job Submission with

slurm

workload manager

Introduction to Slurm

- Simple Linux Utility for Resource Management (SLURM):
  - **Main entities:**
    1. **Nodes:** Physical computing resources
    2. **Partition:** A logical set of nodes
    3. **Jobs:** Allocations of resources assigned to a user for a specified amount of time
    4. **Job Steps:** sets of (possibly parallel) tasks within a job
    5. **Tasks:** Implies the requested/allocated computing resources to process(es) per job or job step
       (By default, each task refers to <u>1 CPU core</u>)



Slurm info Commands

- **`sinfo`**:
  - View information about nodes and partitions. (similar to `qstat -g c` command in UGE)
    - **PARTITION:** The name of the available partitions in the cluster
    - **AVAIL:** shows the current state of the partition: `up`, `down`, `drain`, `inactive`.
      - Make sure the partition is `up` before submit a job
    - **TIMELIMIT:** always shows `infinite`.
      - The time limit per job will be enforced based on the "`account`" you choose for your job.
    - **NODES:** Shows the number of nodes in a particular state.
    - **STATE:** Indicates the state of a group of nodes:
      - `idle`: nodes are available and ready for allocation
      - `mix`: nodes are partially allocated
      - `alloc`: nodes are fully allocated
      - `drain/drang`: nodes are not available but current running jobs will continue until they finish
      - `down/unk`: nodes are down, and no job is running in those nodes
    - **NODELIST:** List of nodes belong to a particular partition/state.

Slurm info Commands

- **`squeue`:**
  - view information about jobs located in partitions. (similar to `qstat` command in UGE)
    - The `squeue` command shows all the users' jobs in all partitions.
    - Useful options to filter the output:
      - **`-u <user>, --user=<user>`:** Shows the list of jobs or job steps that belong to a specific user
      - **`-p <partition>, --partition=<partition>`:** Filters the jobs within a partition.
    - The `squeue` has been configured on the login nodes to show the most useful data. However, users can still modify the format of output by using:
      `-O <output_format>, --Format=<output_format>`
    - For more details, please refer to manual page of `squeue`.

Slurm info Commands

- **`squeue`** (cont.):
  - Command output:
    - **JOBID:** unique id of jobs
    - **PARTITION:** the name of the job’s partition.
    - **PRIORI:** shows the priority of the jobs calculated by fair-share algorithm. Larger the number, sooner the job get allocated.
    - **ST:** states of the jobs: `PD` (pending), `R` (running), `CA` (canceled), `CG` (completing), `F` (Failed)
    - **USER:** the username of the user’s job
    - **NAME:** the name of the job defined by the user
    - **TIME:** the duration of the running job.
    - **NODES:** number of allocated nodes
    - **CPUS:** number of allocated CPU cores
    - **NODELIST(REASON):** the list of allocated nodes if job is running OR the reason the job is in `PD` or `F`.

Job Submission in Slurm

- **`sbatch`**:
  - submits a job script for later execution. (similar to `qsub` command in UGE)
    - The submitted job stays in the queue until the requested resources become available.
    - The job submission script is a text file that contains “`#SBATCH`” hints with `sbatch` command line options

```
#!/bin/bash
#SBATCH -J MPI_test
#SBATCH -N 2
#SBATCH -ntasks-per-node=128
#SBATCH -o %x.%j.out
#SBATCH -e %x.%j.err
#SBATCH -p nocona

module load gcc/10.1.0 openmpi/3.1.6
mpirun ./my_mpi
```

Job Submission in Slurm

- Job Submission Script Layout:

Description	UGE	SLURM
Transfer environment variables to the job env	-V	--export=[ALL | NONE | variables]
Start the command from current working directory	-cwd	Not necessary
Use /bin/bash as the shell	-S /bin/bash	N/A: Slurm uses bash by default
Set the name for job	-N Jobname	-J , --job-name=<jobname>
The name of the standard output file	-o <filename pattern>	-o, --output=<filename pattern>
The name of the standard error file	-e <filename pattern>	-e, --error=<filename pattern>
Define the queue (partition) name	-q <queue name>	-p, --partition=<partition_names>
Type of parallel env for job/task allocation	-pe <parallel env> cores	-N, --nodes=<# of nodes> --ntasks-per-node=<ntasks>
Reserve memory per slot	-l h_vmem=<float>G	--mem-per-cpu=<size[K|M|G|T]>
Set the maximum job run time	-l h_rt = HH:MM:SS	-t, --time=<HH:MM:SS>
Specify the cluster policy for this job	-P <project name>	-A, --account=<account> | -q, --qos

Job Submission in Slurm

- Select a partition:
  - Partition in Slurm groups physical nodes into a logical set and allows jobs to request for nodes' resources from that partitions.
    - `-p, --partition=<partition_name>`

Name	# of Nodes	Type	Nodes	#Core/Node	#Mem/Node	#Mem/Core	#GPU/node
nocona	240	AMD ROME CPU	cpu-[23-26]-[1-60]	128	503 GB	3.9 GB	N/A
matador	20	Intel/Nvidia V100 GPU	gpu-[20-21]-[1-10]	40	376 GB	9.4 GB	2
gpu-build	1	Intel/Nvidia V100 GPU	gpu-20-11	32	187 GB	5.9 GB	1
quanah	467	Intel Xeon Broadwell	cpu-[1-10]-[*]	36	188 GB	5.3 GB	N/A

Name	# of Nodes	Type	Nodes	#Core/Node	#Mem/Node	#Mem/Core	Available
ivy	100	Intel Xeon Ivy Bridge	Cpu-[17-19]-[*]	20	63 GB	3.1 GB	TBA
community clusters	*	*	*	*	*	*	TBA

Job Submission in Slurm

- Requesting CPU:
  - In Slurm, unlike UGE, there is no Parallel Environment (PE). [`-pe mpi 72`]
    Instead, users must define the following options in their job submissions in order to request for CPU resources:

    1. Number of nodes: How many total nodes for the job?
       - **`-N, --nodes=<number of nodes>`**
    2. Number of tasks per node: (*Recommended*) (By default, each task consumes **1x** CPU core)
       - **`--ntasks-per-nodes=<number of task per node>`**

       OR Number of total tasks: How many task across the nodes?
       - **`-n, --ntasks=<number of tasks>`**
    3. Number of cores/threads per task: (*Optional*)
       - `-c, --cpus-per-task=<#cpus>`
       - `--threads-per-core=<#threads>`

Job Submission in Slurm

- Tips and Recommendations:
  - It would be wise to choose the number of nodes and tasks carefully and efficiently:
    - Try to use up all the cores in one node before request for additional nodes, otherwise your job will face with more network/process overhead.
      - `--partition=nocona --nodes=2 --ntasks=32` ✗
      - `--partition=nocona --nodes=1 --ntasks=32` ✓ (e.g. Shared-memory / serial jobs)
      - `--partition=nocona --nodes=2 --ntasks=256` ✓ (e.g. Distributed / MPI jobs)
  - Changing the number of cores per task or number of threads per core will be reflected in total number of requested cores:
    - `--nodes=2 --ntasks-per-node=64 --cpus-per-task=2` ⇨ 2 x 64 =128 core/node
    - --nodes=1 --ntasks=32 --threads-per-core=2 ⇨ 32 x 2 = 64 total cores for this job.
    - The default number of 1 core per task should be preferable for most of the jobs.

Job Submission in Slurm

- Requesting Memory:
  - One can specify the size of the consumable Memory in two ways in Slurm:
    1. Memory per core (*Recommended*):
       - **`--mem-per-cpu=<size[M|G]>`**
    2. Memory per node:
       - **`--mem=<size[M|G]>`**
  - If no memory size was specified, Slurm will assign the default memory per core to your job:

Nocona: 4027 MB (3.9 GB) per core **Matador:** 9639 MB (9.4 GB) per core **Quanah:** 5370 MB (5.3 GB) per core

  - Once specified the memory size for your job, Slurm will allocate the same amount of physical memory (RAM) to the job + 25% swap space on the node(s):
    - `--nodes=1 --ntasks=32 –mem-per-cpu=2GB`
    - **Soft Limit:** 32 x 2GB = 64GB Memory per node (RAM space)
    - **Hard Limit:** 64 GB + (10% of 64GB) = 64 GB RAM + 16 GB Swap = 80GB total Memory

Job Submission in Slurm

- Requesting Runtime limits:
  - Recommended that you set the max runtime you expect a job will take.
    - **`-t, --time=<time>`**
    - `<time>` can be:
      - *minute*
      - *minute:seconds*
      - ***hours:minutes:seconds***
      - *days-hours*
      - *days-hours:minutes*
      - *days-hours:minutes:seconds*
    - `E.g., --time=24:00:00`
  - Please note that there is a 48-hour default time limit per job and exceeding this amount will end up with rejecting your job submission.

Job Submission in Slurm

- Requesting GPU:
  - GPUs are available by requesting any node in the matador partition.
    - Number of GPUs per node (*Recommended*):
      - **`--gpus-per-node=[<type>:]<number>`**
    - Total number of GPUs:
      - **`-G, --gpus=<# of gpus>`**
  - There is only one type of GPU in RedRaider Cluster (`v100`) and is optional to be specified.
  - It is required to requesting at least **one GPU per node** when submitting a job to Matador.
  - Make sure you do not exceed more than 2 GPUs per node during the job submission.
    - `--partition=matador --nodes=2 --gpus=6` ✗
    - `--partition=matador --nodes=2 --gpu-per-node=2` ✓

Job Submission in Slurm

- Choosing an Account:
  - Accounts, in Slurm, assigns the usage/fair-shair policies to each job. (Like `–P project` in UGE)
    - **`-A, --account=<account>`**
  - The "default" account will be assigned to every job by default, unless a different account is specified
  - List of available accounts on RedRaider cluster is shown in the next slide.

- Selecting QoS:
  - QoS in Slurm defines a set of pre-defined resource limits based on the selected account.
    - **`-q, --qos=<QoS>`**
  - Each account on RedRaider has a default QoS that will be assigned to every job by default.
  - A non-default QoS must be defined explicitly in job submissions to be applied to the job.
  - List of available QoSs for each account on RedRaider cluster is shown in the next slide

Job Submission in Slurm

- List of Accounts/QoS on RedRaider Cluster:

Account -A, --account	**QoS** -q, --qos	**Default Runtime**	**Maximum Runtime**	**Total CPU/Mem Limits**	**CPU/Mem Limit per job**	**Allowed Partitions**	**Priority**
default *	**normal ***	**48 hours**	**48 hours**	**No limit**	**No limit**	**All Partitions**	**normal**
	xlquanah	72 hours	120 hours	144 cores / 755GB	36 cores / 188GB	quanah	normal
aquino, herrera, jiao, lin	aquino*, herrera*, jiao*, lin*	72 hours	No limit	Varies based on the purchased resources	Up to the total available resources	nocona	high
hep, cbg	hep*,cbg*	72 hours	No limit	Varies based on the purchased resources	Up to the total available resources	quanah	high

(*) System will assign the default Account/QoS if user does not define them in their job submissions.

Job Submission in Slurm

- **Account and QoS Examples:**

1. A normal user with `default` account requests for `xlquanah` on `quanah` partition with 5 days runtime limit.
2. A member of Dr. Aquino’s group requests for `aquino` account on `nocona` partition with 10 days runtime limit.

(1)

```
#!/bin/bash
#SBATCH -J MPI_test
#SBATCH -N 1
#SBATCH -ntasks=128
#SBATCH -o %x.%j.out
#SBATCH -e %x.%j.err
#SBATCH -p quanah
#SBATCH -q xlquanah
#SBATCH -t 120:00:00
```

(2)

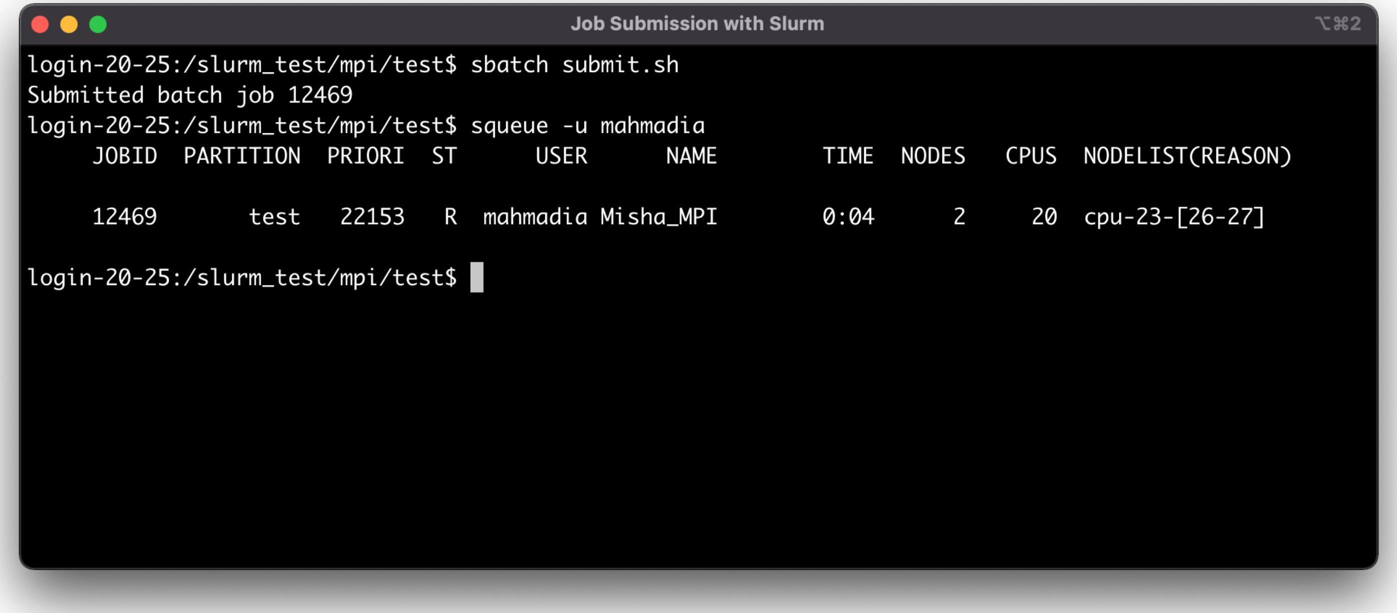
```
#!/bin/bash
#SBATCH -J MPI_test
#SBATCH -N 2
#SBATCH -ntasks-per-node=128
#SBATCH -o %x.%j.out
#SBATCH -e %x.%j.err
#SBATCH -p Nocona
#SBATCH -A aquino
#SBATCH -t 10-00:00:00
```

Job Submission in Slurm

- Submit a job to Slurm:
  - Create a job submission script file (e.g., submit.sh):

```
#!/bin/bash
#SBATCH -J MPI_test
#SBATCH -N 2
#SBATCH -ntasks-per-node=128
#SBATCH -o %x.%j.out
#SBATCH -e %x.%j.err
#SBATCH -p nocona

module load gcc/10.1.0 openmpi/3.1.6
mpirun ./my_mpi
```

  - Submit the job with `sbatch`:
    - **`sbatch submit.sh`**
  - Monitor the job with `squeue`:
    - **`squeue -u <username>`**
  - Cancel the job with `scancel`:
    - **`scancel job_id`**



```
login-20-25:/slurm_test/mpi/test$ sbatch submit.sh
Submitted batch job 12469
login-20-25:/slurm_test/mpi/test$ squeue -u mahmadia
     JOBID  PARTITION  PRIORI  ST      USER       NAME         TIME  NODES    CPUS  NODELIST(REASON)

     12469       test   22153   R  mahmadia Misha_MPI          0:04      2      20  cpu-23-[26-27]

login-20-25:/slurm_test/mpi/test$ █
```

Job Submission in Slurm

- **srun**:
  - submits a job for execution or initiates job steps in real time.
  - `srun` has the same options as `sbatch` with a few more. (Please see the man page)
  - `srun` works similar to the "`mpirun`" and it can be replaced with "`mpirun`" as well.

```
#!/bin/bash
#SBATCH -J MPI_test
#SBATCH -N 2
#SBATCH -ntasks-per-node=128
#SBATCH -o %x.%j.out
#SBATCH -e %x.%j.err
#SBATCH -p nocona

module load gcc/10.1.0 openmpi/3.1.6
mpirun ./my_mpi
```

```
#!/bin/bash
#SBATCH -J MPI_test
#SBATCH -N 2
#SBATCH -ntasks-per-node=128
#SBATCH -o %x.%j.out
#SBATCH -e %x.%j.err
#SBATCH -p nocona

module load gcc/10.1.0 openmpi/3.1.6
srun ./my_mpi
```

Job Submission in Slurm

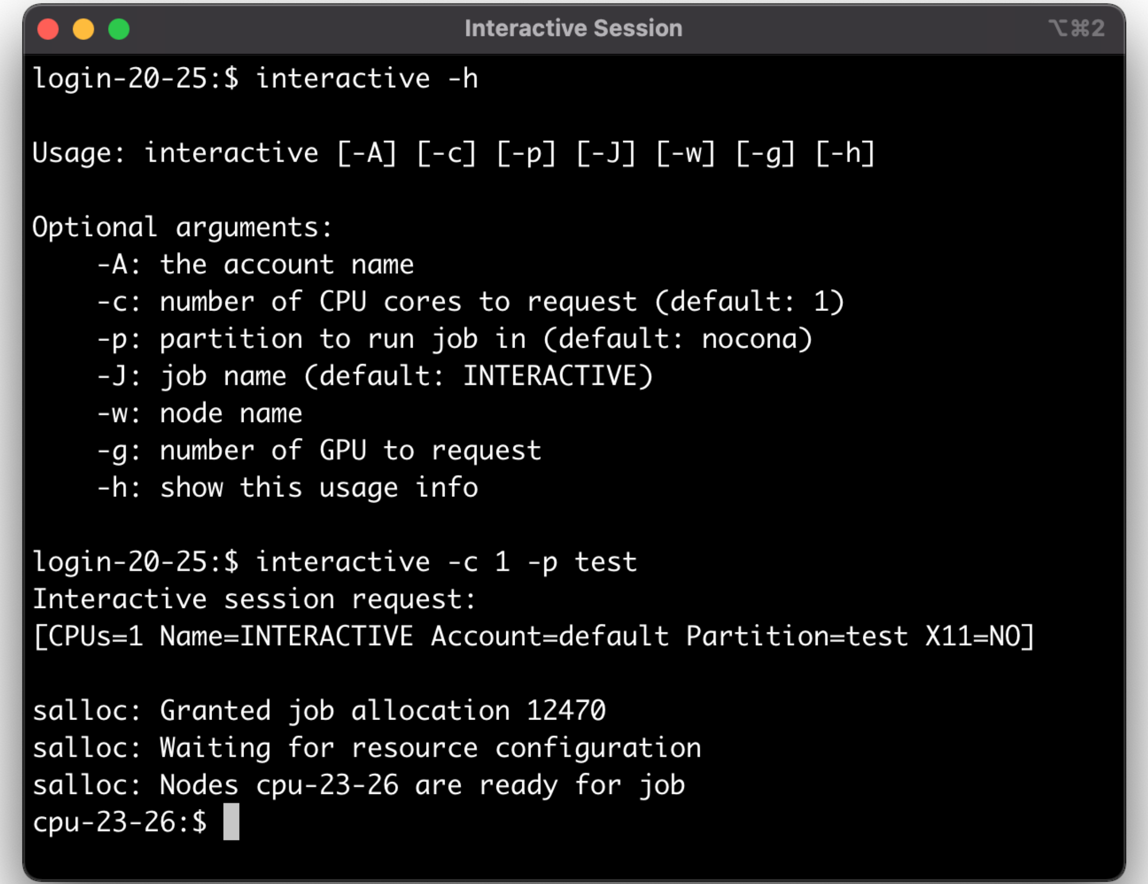
- **srun**:
  - srun can launch any non-distributed (serial/multi-threaded) processes as well.
  - Multiple programs can be launched by srun with different CPU/Mem size within an allocated job.

```
#!/bin/bash
#SBATCH -J MPI_test
#SBATCH -N 1
#SBATCH -ntasks=1
#SBATCH -o %x.%j.out
#SBATCH -e %x.%j.err
#SBATCH -p nocona


srun ./my_serial_prog.exe
```

```
#!/bin/bash
#SBATCH -J MPI_test
#SBATCH -N 3
#SBATCH -ntasks-per-node=128
#SBATCH -o %x.%j.out
#SBATCH -e %x.%j.err
#SBATCH -p nocona

srun -N 1 --ntask=128 ./my_sm_app1 &
srun -N 1 --ntask=128 ./my_sm_app2 &
srun -N 1 --ntask=128 ./my_sm_app3
```

Interactive Session

- **`interactive`**:
  - Starts an interactive session/job (similar to `qlogin`):
    - `interactive -c 2 -p nocona`
    - See the `interactive –h` for all the available options.
  - Make sure the prompt changes to `cpu-#-#`.
  - Make sure you run “exit” when you’re finished.
  - Keep in mind resource/runtime limits apply to `interactive` based on the selected account.
  - The `interactive` command will forward the X11 if the SSH session was established with `–X` or `–Y`.
  - Please note that direct SSH to the nodes is blocked on RedRaider cluster.



```
login-20-25:$ interactive -h

Usage: interactive [-A] [-c] [-p] [-J] [-w] [-g] [-h]

Optional arguments:
    -A: the account name
    -c: number of CPU cores to request (default: 1)
    -p: partition to run job in (default: nocona)
    -J: job name (default: INTERACTIVE)
    -w: node name
    -g: number of GPU to request
    -h: show this usage info

login-20-25:$ interactive -c 1 -p test
Interactive session request:
[CPUs=1 Name=INTERACTIVE Account=default Partition=test X11=NO]

salloc: Granted job allocation 12470
salloc: Waiting for resource configuration
salloc: Nodes cpu-23-26 are ready for job
cpu-23-26:$ █
```

The ‘gpu-build’ Partition

- Building and Testing GPU applications:
  - The `gpu-build` partition contains one Intel/GPU node with **1x Nvidia V100** GPU device, **32x Intel CPU cores** and **192 GB RAM**, which allows users to:
    - Build their own GPU applications.
    - Test GPU applications and the environment setup before submitting a job to Matador partition.
    - Accessing the Lmod Module environment for GPU compilers/applications.
  - In order to access the ‘gpu-build’ node, you need to establish and `interactive` session:
    - **`$ interactive -p gpu-build -c 2`**
  - Limitations:

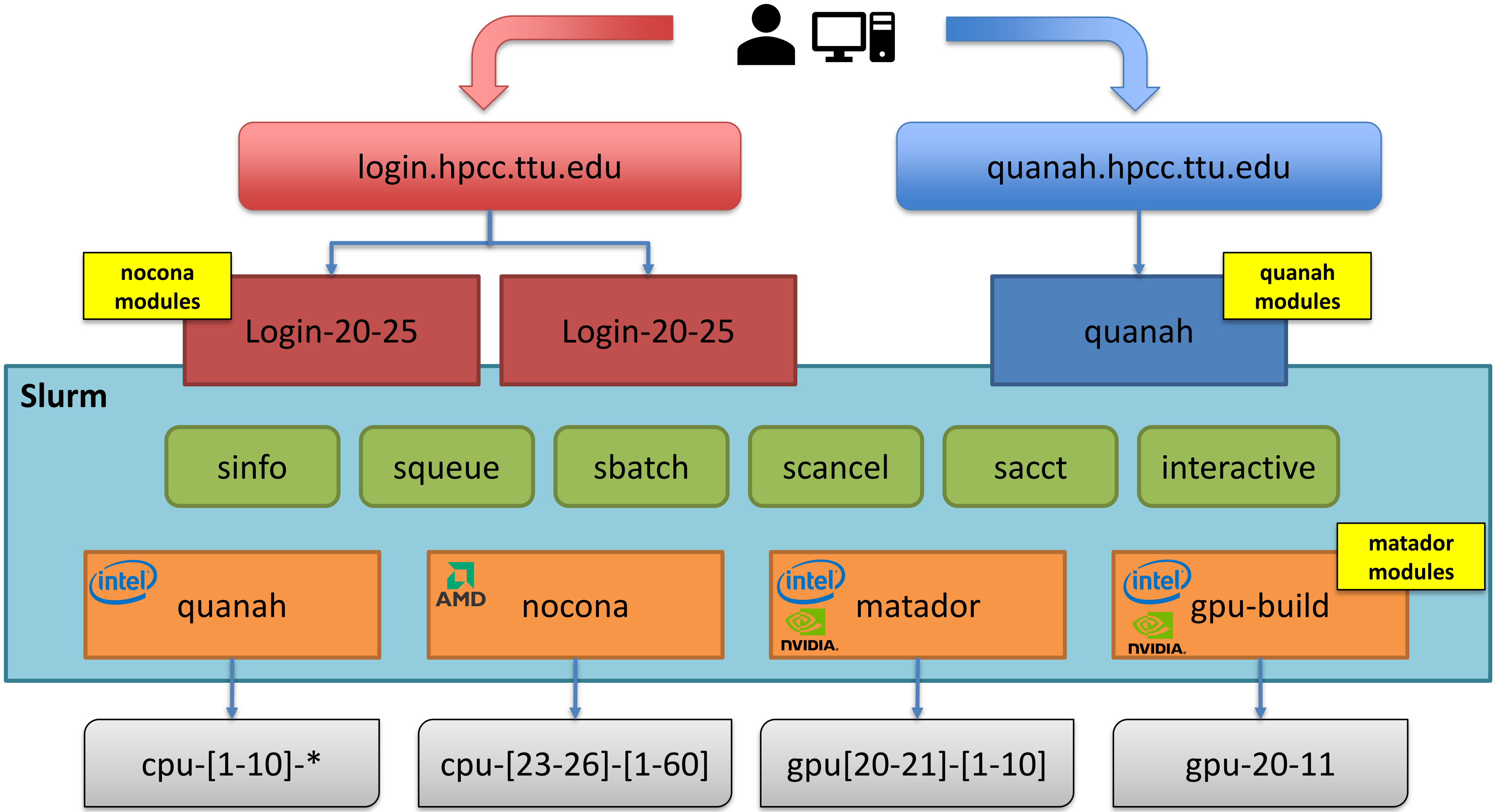
Partition	Max Runtime (per job)	Max CPU per user (in total)	Max Mem per user (in total)	Max interactive session per user
gpu-build	5 hours	6	36006 MB (35 GB)	2

Debugging a Finished Job

- **`sacct`:**
  - reports accounting information about active or completed jobs or job steps.
    - **`sacct -j <jobid>`**
  - More filter options are available by checking the `–e, --helpformat` options of `sacct` command.
    - `sacct -j <jobid> --format=partition,jobid,ntasks,nodelist,maxrss,maxvmsize,exitcode`
  - When debugging:
    - Check the output and error files
    - Check the output of sacct for:
      - ✓ Memory usage
      - ✓ Exit code
      - ✓ Start and end time.

Software builds and installation

HPCC RedRaider Cluster – Overall Look



HPCC RedRaider Cluster - CPU Architectures

- Multiple partitions – Multiple architectures:

Nocona

AMD EPYC ROME

Matador

Intel Xeon Cascade Lake
Nvidia V100

Quanah

Intel Xeon Broadwell

Ivy

Intel Xeon Ivy Bridge

Software builds on HPCC Clusters

- What that means?
  - Each CPU architecture may bring a different set of features and instructions.
  - Compiled programs (C/C++/Fortran) need to be re-compiled against each CPU architecture.
  - E.g., programs that are compiled on **Intel** nodes may not work properly/efficiently on **AMD** nodes.
  - Different Compilers and Math libraries optimize the programs in different ways on various archs:

Compiler	AMD ROME	Intel Broadwell	Intel Ivy Bridge	Intel Cascade Lake	Nvidia V100
GNU/GCC	**GCC 10+**	GCC 4+	GCC 4+	**GCC 10+**	GCC 8+
Intel	Not optimized	**Optimized**	**Optimized**	**Optimized**	Intel 19+
AOCC	**Optimized**	Not Applicable	Not Applicable	Not Applicable	N/A
MKL	Not optimized	Optimized	Optimized	Optimized	MKL 19+
AOCL	**Optimized**	Not Applicable	Not Applicable	Not Applicable	N/A
CUDA	N/A	N/A	N/A	N/A	**CUDA 10+**

Software builds on HPCC Clusters

- Tips and Recommendations:

1. Create a separate directory for each CPU architecture, and make a copy from your code/program and place it under each directory:

   ```
   mkdir nocona matador quanah
   ```

2. Login to the RedRaider login node, and for each CPU architecture make an interactive session to the corresponding worker node:

   ```
   interactive -p nocona -c 10
   ```

3. Go to the directory of you code that has the same name as the current session's partition:

   ```
   cd nocona
   ```

4. Load a proper compiler module and recompile your code:

   ```
   module load gcc/10.1.0
   ```

5. If applicable, add the `-O3` optimization flag to all the `CFLAGS`, `CPPFLAGS`, `CXXFLAGS`, `FFLAGS`.

   ```
   CFLAGS=-O3 FFLAGS=-O3 make -j 10 all
   ```

Software builds on HPCC Clusters

- Tips and Recommendations:

5. We recommend mapping the MPI jobs to the L3-cache memory on **Nocona (AMD)** nodes:
   - `mpirun -map-by l3cache -bind-to core ./mpi_app`

6. **HPCC will not support Python v2 on Nocona and Matador nodes with CentOS 8. (This rule will be applied to Quanah and Ivy in the near future.)**
   - Users can still get Python v2 from Conda (Anaconda/Miniconda)
   - Python 2 is NOT RECEIVING SECURITY UPDATES and should be retired from your workflows ASAP.

7. Python applications (including the applications from Condo repo) will continue working with different architectures without recompiling them.

8. The pre-installed version of CUDA can be found under this directory on Matador nodes:
   - `/usr/local/cuda`

Getting Help

Further Assistance

- Visit Our Website:
  - Most user guides have been updated
  - New user guides are being added
  - https://www.depts.ttu.edu/hpcc/userguides/index.php
- Read the documentation!
  - https://slurm.schedmd.com/documentation.html
- Submit a support ticket:
  - Send an email to hpccsupport@ttu.edu

TEXAS TECH UNIVERSITY

Information Technology Division™