Introduction to High-Performance computing (HPC)

Training Slides Created & Maintained by:

Yue Yu

Sr. Research Computing Facilitator of Cyberinfrastructure and Research Technologies (CIRT)

Office of Information Technology (OIT) University of California Merced

Training Presented by:

Sarvani Chadalapaka

Director of Cyberinfrastructure and Research Technologies (CIRT)

Alex Villa

Student Tech Consultant

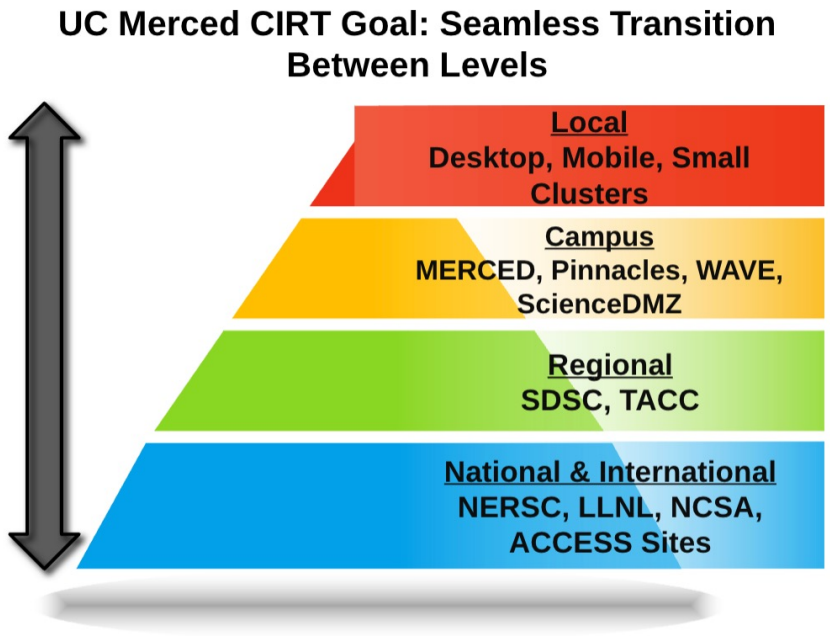
Office of Information Technology (OIT)

University of California Merced

Today's AGENDA

- What is Supercomputing(HPC)? (10 minutes)
  - Priority 1: Supercomputing terminology
  - Priority 2: Supercomputing at University of California, Merced
- Terminology and HPC Architecture(5 minutes)
- HPC Clusters at UC Merced (5 minutes)
  - MERCED Cluster
  - Pinnacles Cluster
- What is a Scheduler and Why it is Used?(5 minutes)
  - Slurm Scheduler
- ~~Resource Queues and Limitations (10 Minutes)~~
- Hands On Training and Q&A(Until end of Training)

CIRT - Goal


UC Merced CIRT Goal: Seamless Transition Between Levels
Local
Desktop, Mobile, Small Clusters
Campus
MERCED, Pinnacles, WAVE, ScienceDMZ
Regional
SDSC, TACC
National & International
NERSC, LLNL, NCSA, ACCESS Sites

What is Supercomputing?

- Supercomputing is the biggest, fastest computing right this minute
- A supercomputer is one of the biggest, fastest computers right this minute
  - So, the definition of supercomputing is constantly changing.

Rule of Thumb: A supercomputer is typically at least 100 times as powerful as a PC.

Jargon:

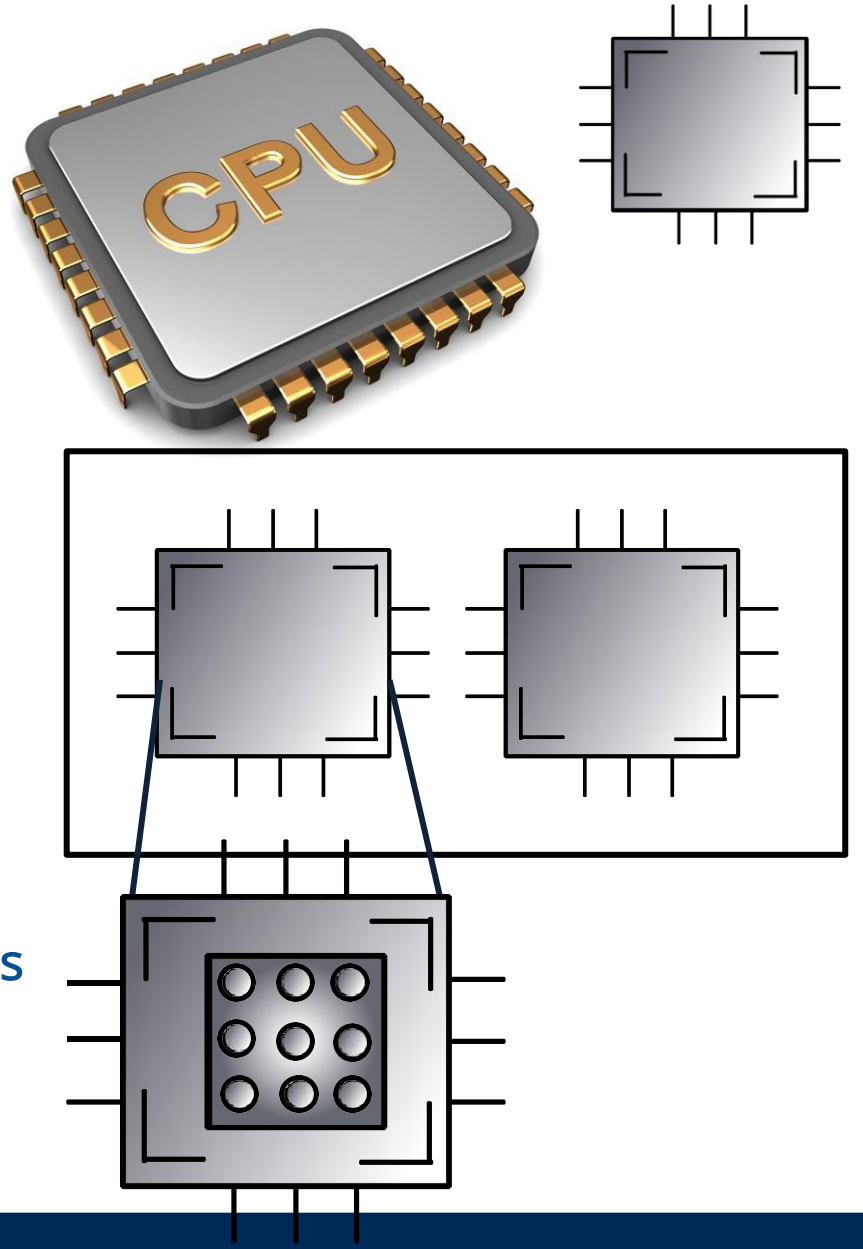
Supercomputing is also known as
High Performance Computing (HPC) or
High End Computing (HEC) or
Cyberinfrastructure (CI)

I am working at Cyberinfrastructure and Research Technologies (CIRT) team

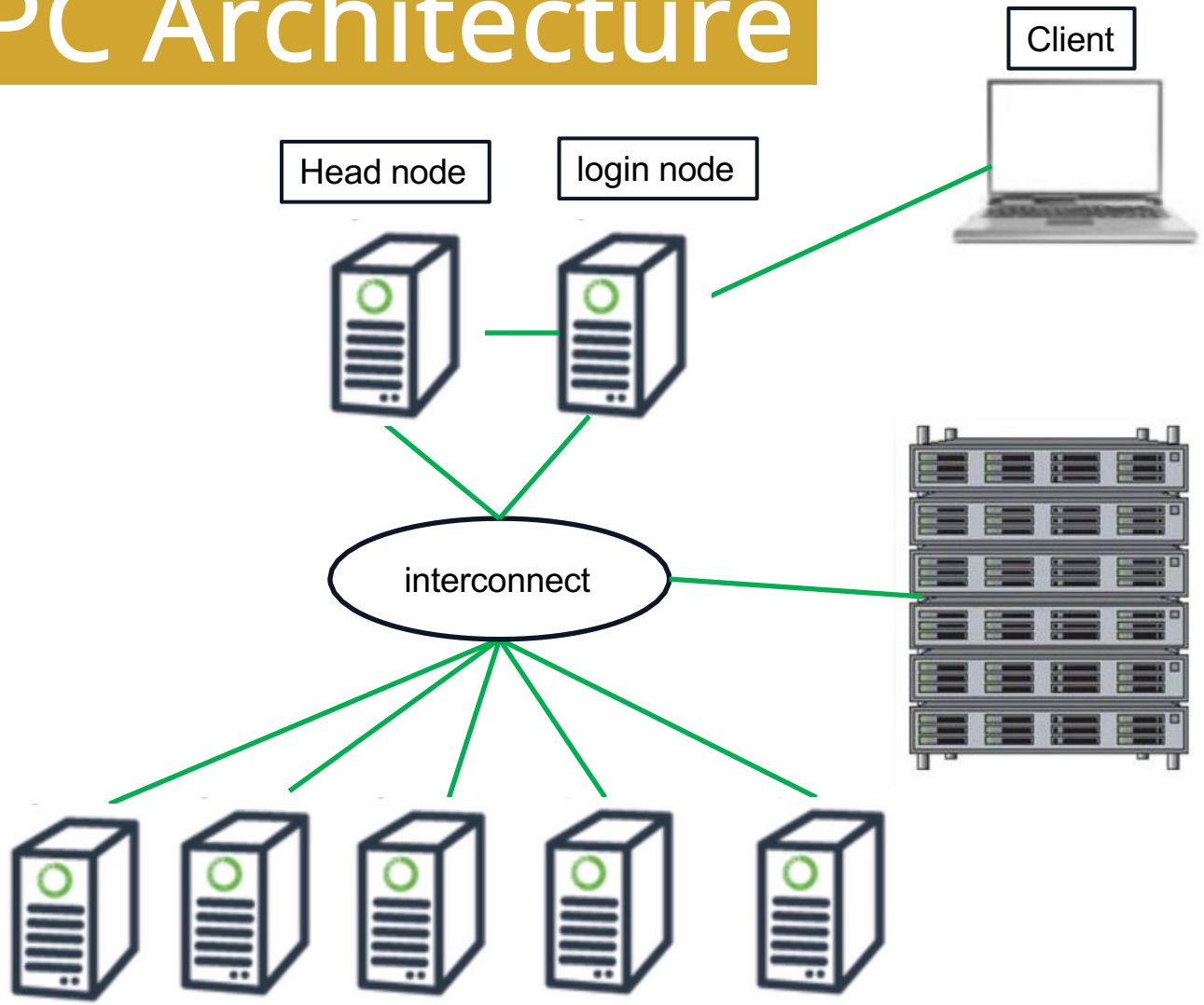
https://it.ucmerced.edu/CIRT

Terminologies

- CPU (processor)
  - Central Processing Unit
- GPU
  - Graphics Processing Unit
  - Deep learning, massive parallelism, 3D rendering…
- Nodes
  - Multiple CPUs
  - CPU nodes
  - GPU nodes
- Cores
  - Processing element
  - 1 CPU may contain multiple cores
  - 1 GPU many smaller specialized cores



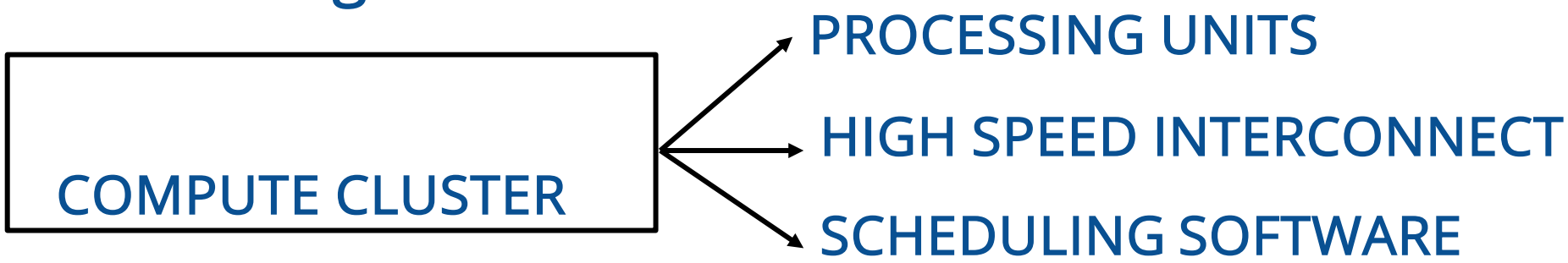
HPC Architecture



- Head node and login node
- Compute nodes (majority of computations are performed)
  - CPU nodes
  - GPU nodes
- Infiniband switches connect all nodes

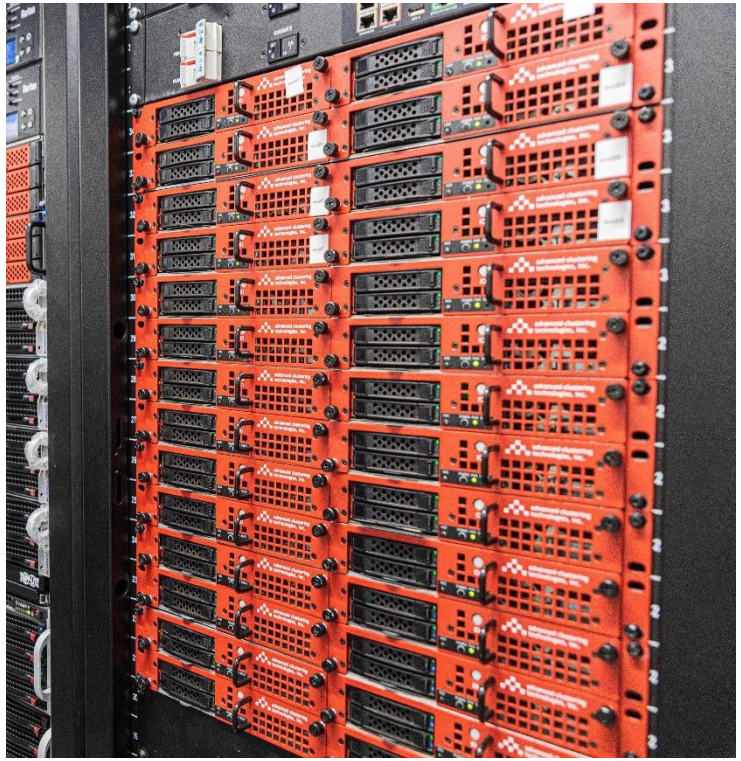
HPC Architecture

- Machines with large number of CPUs and memory
- High-speed interconnect
- Scheduling software



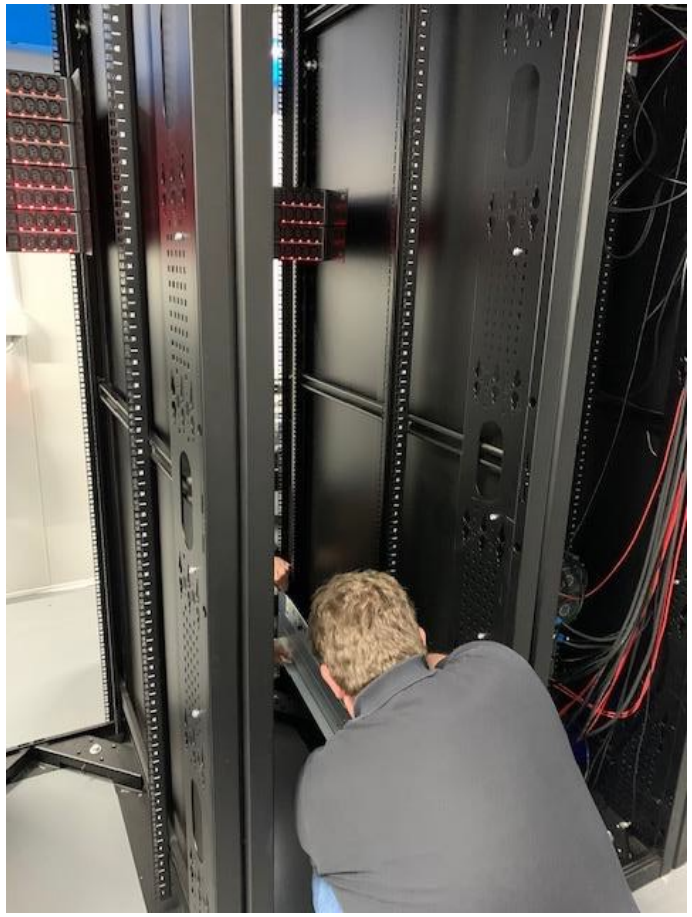
MERCED Cluster Physical Setup

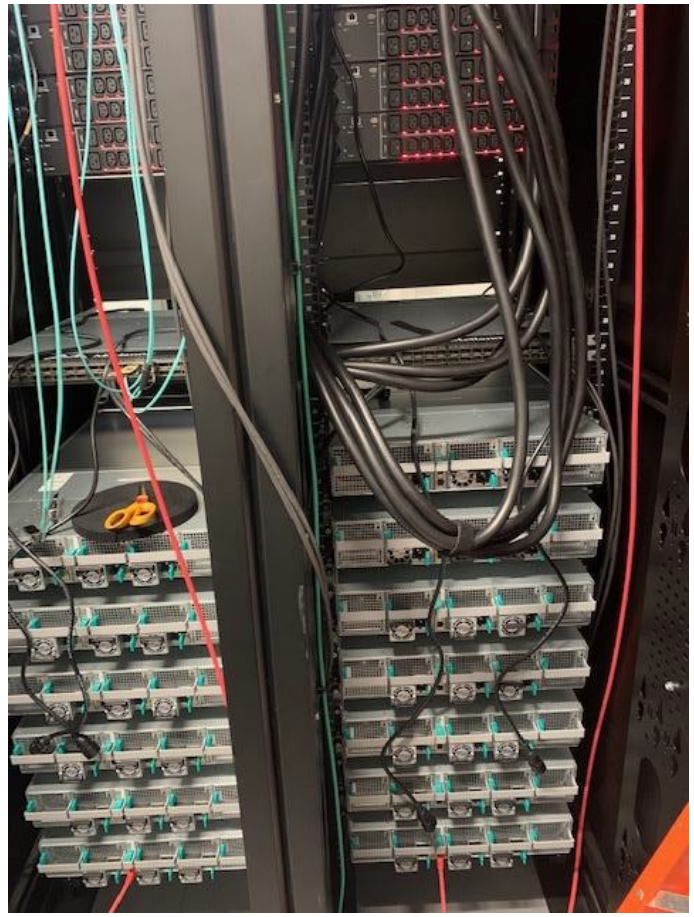
- MERCED cluster is located at Borg Cube



- Head node + Login node + 80 CPU compute nodes
  - Multigenerational nodes
    - Haswell, Broadwell, Skylake…
    - 128GB/256GB of RAM
- Cluster storages + 786TB Borg storage + PI owned Storage units

Pinnacles Installation





Pinnacles Cluster

- Head + 2 login nodes
- 40 CPU compute nodes
- 4 big memory CPU nodes
- 8 GPU nodes

Infiniband Architecture

- InfiniBand (IB) networking communications
- RDMA (Remote Direct Memory Access)

- 10GigE has 5-6 times the latency of IB

- IB has 3.7x the throughput of 10GigE

Shared Filesystem

Mounted across the cluster –

Centralized Login

Folder	Space
/home/<UCMID>/data	500G
/home/<UCMID>/scratch	500G
/home/<UCMID>/	70G

Scheduler

- Scheduling is the method by which work specified by some means is assigned to resources that complete the work. The work may be virtual computation elements such as threads, processes or data flows, which are in turn scheduled onto hardware resources such as processors, network links or expansion cards.

- A scheduler is what carries out the scheduling activity. Schedulers are often implemented so they keep all computer resources busy (as in load balancing), allow multiple users to share system resources effectively, or to achieve a target quality of service.

Slurm Scheduler

Slurm has three key functions.

1. Allocates exclusive and/or non-exclusive access to resources (compute nodes) to users for some duration of time so they can perform work.
2. Provides a framework for starting, executing, and monitoring work (normally a parallel job) on the set of allocated nodes.
3. Arbitrates contention for resources by managing a queue of pending work.

Queues & Resource Limitations

- Currently, priority on the queues is setup such that each project is given equal priority (each project is assigned to a PI) and each member in that project (or PI group) has equal priority.
- To view the queues and note information
  - Type “sinfo” in your terminal

```
[root@rclogin01 ~]# sinfo
CLUSTER: pinnacles
PARTITION         AVAIL  TIMELIMIT  NODES  STATE NODELIST
test                 up    1:00:00     24    mix gnode[009-010,012-013],hmnode[001,003-004],node[001-003,005-007,016,
019-021,026,030,034,037,045,053,063]
test                 up    1:00:00     29  alloc gnode[001-003,005],hmnode[002,005],node[004,008,014-015,017-018,022,
024-025,027-029,032-033,035,039,041-044,055-057]
test                 up    1:00:00     24   idle gnode[004,006-008,011],node[013,023,031,036,038,040,046-052,054,058-
062]
bigmem               up 3-00:00:00      3    mix hmnode[001,003-004]
bigmem               up 3-00:00:00      1  alloc hmnode002
gpu                  up 3-00:00:00      4  alloc gnode[001-003,005]
gpu                  up 3-00:00:00      4   idle gnode[004,006-008]
short*               up    6:00:00     14    mix node[001-003,005-007,016,019-021,026,030,034,037]
short*               up    6:00:00     16  alloc node[004,008,014-015,017-018,022,024-025,027-029,032-033,035,039]
short*               up    6:00:00      6   idle node[013,023,031,036,038,040]
medium               up 1-00:00:00     14    mix node[001-003,005-007,016,019-021,026,030,034,037]
medium               up 1-00:00:00     16  alloc node[004,008,014-015,017-018,022,024-025,027-029,032-033,035,039]
medium               up 1-00:00:00      6   idle node[013,023,031,036,038,040]
long                 up 3-00:00:00     14    mix node[001-003,005-007,016,019-021,026,030,034,037]
long                 up 3-00:00:00     16  alloc node[004,008,014-015,017-018,022,024-025,027-029,032-033,035,039]
long                 up 3-00:00:00      6   idle node[013,023,031,036,038,040]
pi.larsson           up   infinite      5  alloc hmnode005,node[041-044]
pi.anierenberg       up   infinite      4   idle node[049-052]
pi.dstrubbe          up   infinite      1    mix node053
pi.dstrubbe          up   infinite      1   idle node054
pi.apribram-jones    up   infinite      1    mix node053
pi.apribram-jones    up   infinite      1   idle node054
dept.appliedmath     up 7-00:00:00      3    mix gnode[009-010],node045
dept.appliedmath     up 7-00:00:00      3   idle node[046-048]
pi.amartini          up   infinite      2  alloc node[055-056]
dept.physics         up 7-00:00:00      1  alloc node057
dept.physics         up 7-00:00:00      1   idle node058
grp.ccbm             up 7-00:00:00      4   idle node[059-062]
dept.cogsci          up 7-00:00:00      1    mix node063
pi.ckim103           up   infinite      1   idle gnode011
pi.bdutagaci         up   infinite      2    mix gnode[012-013]
[root@rclogin01 ~]#
```

Documentation website

- https://ucmerced.github.io/hpc_docs/#/README

Guest Account Login: (During the session you will be given a guest login information)

Otherwise, you can use your own account for this practice session

1. ssh Your_Acct_Name@login.rc.ucmerced.edu
    1. “ucm_Pinnacles” - Password
2. To see the available modules that are installed use the following:

    module avail

3. Copy the practice files: (always use the Tab key from keyboard to help you for autocompletion)

    cp -r /home/yyu49/hpc_training/script/serial .

    cp -r /home/yyu49/hpc_training/script/parallel .

The two folders have some sample scripts, user can play around with them

4. Check job: squeue --me

Job Submission Script

```
#! /bin/bash  ➔ "The hashbang line"
#SBATCH  --nodes=1
#SBATCH  --ntasks=1
#SBATCH -p test
#SBATCH --time=0-00:15:00      # 15 minutes
#SBATCH --output=my_%j.stdout
#SBATCH --job-name=test
#SBATCH --export=ALL
whoami
```

How do I actually submit the job?

```
sbatch sample.sub
```

Simple Job Submission Demo

- Run single python job

  Python_test1.py

- Check the status of the job using “squeue –u username” or “squeue --me” commands

Getting Help

Troubleshooting a Job:
https://ucmerced.github.io/hpc_docs/#/Manage_job

Requesting Support Via ServiceNow Ticket System:
https://ucmerced.service-now.com/servicehub?id=sh_new

HPC Office Hours

WHERE? Online & In-person(ACS 312)

WHEN? Every Friday from 10:30 am – 12 noon *

Login page

CIRT website

Documentation page

Additional Resources

- Slurm overview - https://slurm.schedmd.com/documentation.html
- Slurm sbatch - https://slurm.schedmd.com/sbatch.html
- Slurm sinfo - https://slurm.schedmd.com/sinfo.html
- Slurm squeue - https://slurm.schedmd.com/squeue.html
- Requesting help from CIRT - https://it.ucmerced.edu/services?field_service_service_catalog_tid=5
- HPC Documentation https://github.com/ucmerced/merced-cluster/wiki/

Diverse Research Groups on Campus

- Natural Sciences – Soil Biogeochemistry, Biological Physics Theory and Computation, Theoretical Atomic and Molecular Physics, Applied Mathematics, Quantum Chemistry, Quantitative Systems Biology

- Engineering – Tribology, Machine Learning, Fault tolerance/resilience in
  large-scale parallel and distributed systems, power-aware computing

- Social Sciences Humanities and Arts- Evolution of Communication, Neural Networks, Vocal Motor Control, Mesoamerican Indigenous literatures and cultures, Central American and Latina/o cultural studies


THE · UNIVERSITY · OF · CALIFORNIA · MERCED ·
LET THERE BE LIGHT
2005