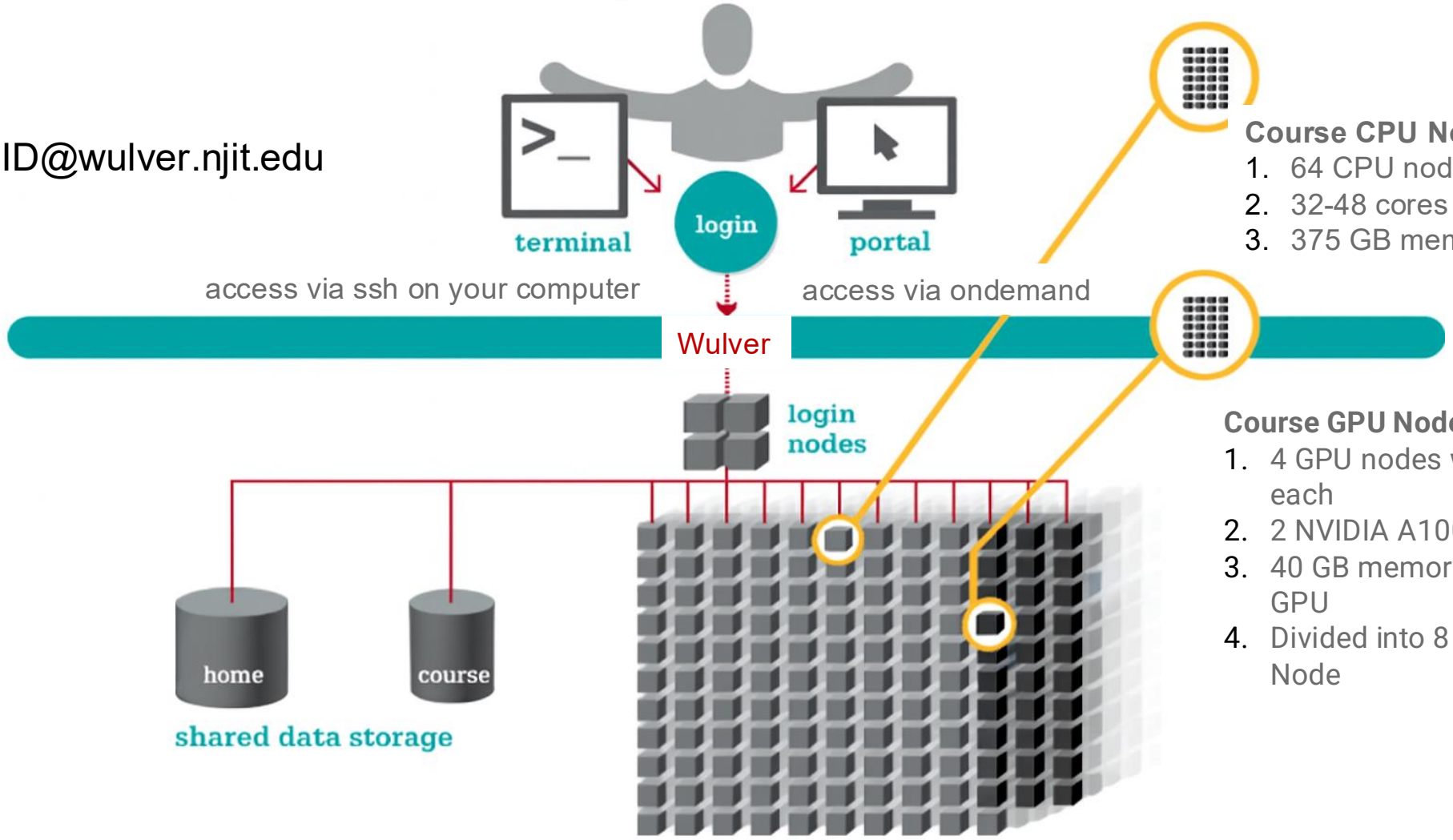
NJIT
New Jersey Institute of Technology

High Performance Computing

Wulver Cluster Specifications

ssh $UCID@wulver.njit.edu



Course CPU Node Specifications

1. 64 CPU nodes
2. 32-48 cores per node
3. 375 GB memory

Course GPU Node Specifications

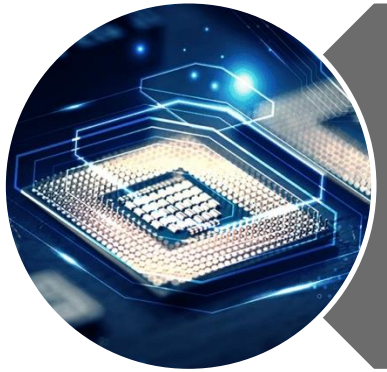
1. 4 GPU nodes with 32-48 cores each
2. 2 NVIDIA A100 GPUs per node
3. 40 GB memory available per GPU
4. Divided into 8 10G MIGs per Node

hpc.njit.edu/clusters/get_started_on_Wulver

HPC Allocations

Storage

- Home (~50GB/user) - Limited quota: not intended for primary storage
- Course (`/course/2026`) - Primary storage

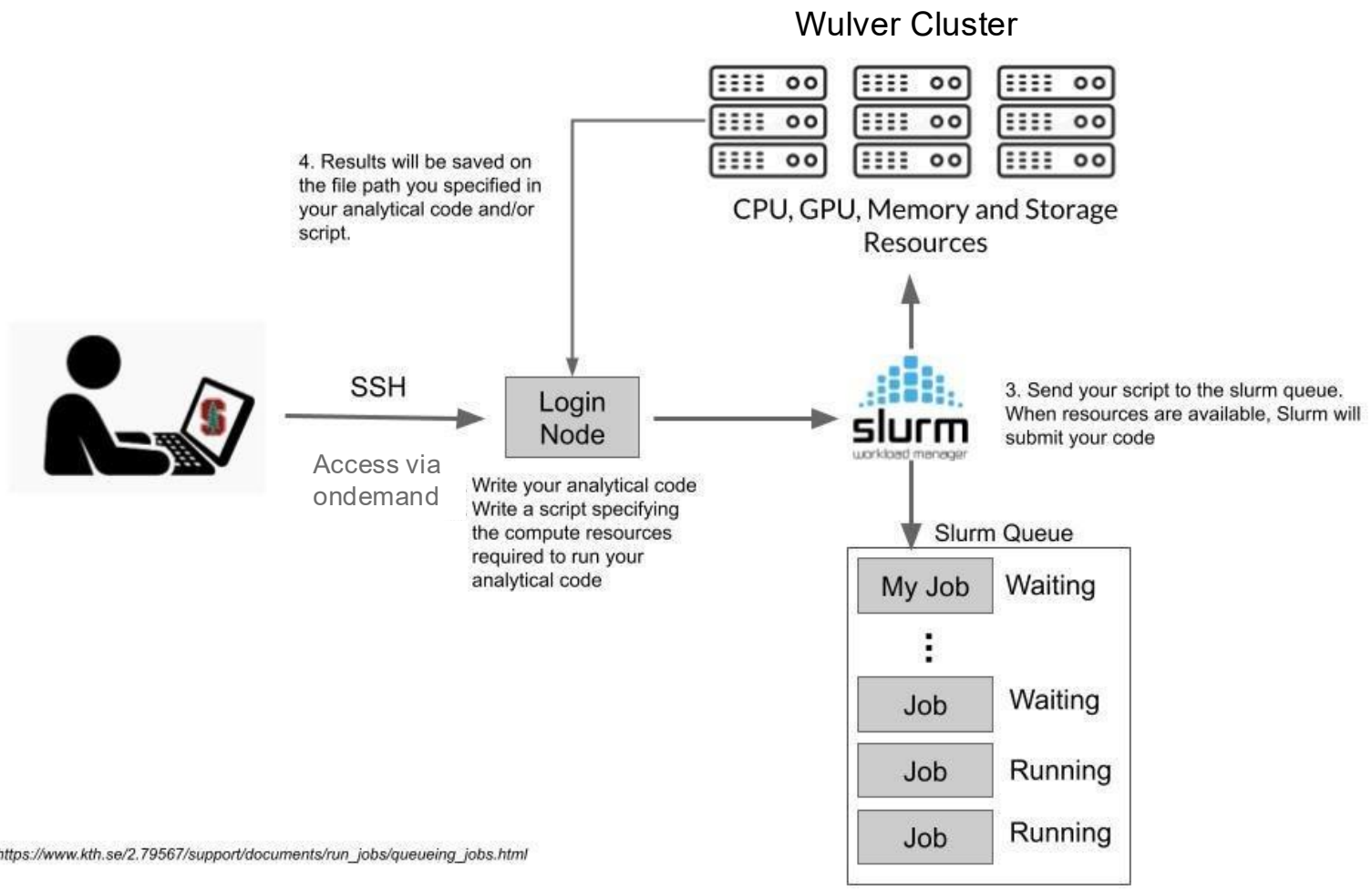


Computing Time

- 1 SU = Number of CPUs x Walltime in hours x usage factor
- SU allocation – 2500 SUs per user

Use `quota_info` to check the allocations

Why do supercomputers use queuing?



Source: https://www.kth.se/2.79567/support/documents/run_jobs/queueing_jobs.html

Manage Jobs – Options

Mandatory Options

Directive	Options	Description
--account=	account	Check your email or use **quota_info**
--partition=	Partition	Request a partition of resources for job allocation (queue)
--time=	Time [[d-]h:]m[:s]	Minimum time limit on job allocation
--qos=	Job Priorities	Define the job priority. Use **--qos=course**

see https://slurm.schedmd.com/srun.html for more details

HPC Partitions

- Example of SU charges: (20 cores with 1 MIG for 8 hours without `--mem`)
  - SU = (20 + 2) * 8 = 176
- Example of SU charges: (20 cores with 1 MIG for 8 hours with `--mem` = 128G)
  - SU = [2 + MAX(20,128/4=32)]*8 = (2 + 32) * 8 = 34 * 8 = 272

Partition	Nodes	Cores/ Node	GPU	Memory	SU charge
`--partition=course`	64	32-48	NA	375 GB	MAX(CPUs, Memory/4G) SU
`--partition=course_gpu`	4	32-48	A100 10G MIG	375 GB	2 + MAX(CPUs, Memory/4G) SU

Job Submission Time Interval Formats

- Valid time formats (with a few exceptions) for `-t / --time=` option

Minutes	(`-t 10` is 10 minutes)
Minutes:Seconds	(…`10:30` is 10 minutes & 30 secs)
Hours:Minutes:Seconds	(`1:0:0` is 1 hr + 0mins + 0secs)
Days-Hours:Minutes:Seconds	(`7-1:10:30` is 7days + 1hr + 10mins + 30secs)
Days-Hours	(`7-0` is 7days + 0hrs i.e. 7 days)
Days-Hours:Minutes	(`7-4:10` is 7days + 4hrs + 10mins)

- Wall time maximum - 72 hours

Manage Jobs – Options

Additional Options

Directive	Options	Description
--ntasks=	Number of cpus	Number of CPUs (tasks) to be allocated
--mem=	Memory	Total memory of the job
--gres=	Generic resources	Set the Number of gpus
--error=	File	Define standard error file
--out=	File	Define standard output file

Directive	Options	Description
--input=	File	File used for standard input
--job	Name	Define job name
--mail-type=	ALL, BEGIN, END, FAIL, REQUEUE	Notify user by email when <type> event occurs
--mail-user=	Email address	Send email to this address for events specified with mail-type option (default is submitting user).

see https://slurm.schedmd.com/srun.html for more details

General Application Workflow

- Log into cluster with ucid and password
- Copy input files to new directory
- Change to copied directory via command line

  `cd /path/to/copied_directory`
- Copy job a template to the directory

  `cp /path/to/templates/`***`jobtemplate`***`.job` **`jobfile`**`.job`
- Modify the job file:
  - Change the number of resources to desired number
  - Change the module load command based on the application name and version
  - Update command line with commands required for job
  - Update the software modules
- Submit the job file using `sbatch`

Sample Simple Job Script

```
#!/bin/bash

#SBATCH --job-name=my_job
#SBATCH --partition=course
#SBATCH --output=%x.%j.out
#SBATCH --error=%x.%j.err
#SBATCH --account=ACCOUNT
#SBATCH --qos=course
#SBATCH --time=00:20:00
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --mail-type=ALL
#SBATCH –mail-user=ab1234@njit.edu

date
sleep 60
date
```

Job setup information for SLURM

Commands to be run

- This runs a batch job called “my_job” to the “course” partition, with 1 task, a wall time limit of 20 minutes.
- QOS is required. Account is recommended.
- `%x.%j` expands to JobName.JobID and prints this into a text file

Check sample job scripts in `/apps/testjobs/`

Sample MPI Job script

```
#!/bin/bash

#SBATCH --job-name=mpi_test_job
#SBATCH --partition=course
#SBATCH --output=%x.%j.out
#SBATCH --error=%x.%j.err
#SBATCH --account=ACCOUNT
#SBATCH --qos=course
#SBATCH --time=00:10:00
#SBATCH --ntasks=64

# Run application commands
srun /apps/testjobs/bin/mpihello
```

- This runs an MPI job named “mpi_test_job”, with 64 processes total. Wall time is 10 minutes.

Check sample job scripts in `/apps/testjobs/workshop`

Sample GPU Job script

```
#!/bin/bash

#SBATCH --job-name=test_gpu_job
#SBATCH --output=%x.%j.out
#SBATCH --error=%x.%j.err
#SBATCH --partition=course_gpu
#SBATCH --account=PI_UCID
#SBATCH --qos=standard
#SBATCH --time=00:20:00
#SBATCH --ntasks=2
#SBATCH --gres=gpu:a100_10g:1

# Load application environment
module load CUDA

# Run application commands
nvidia-smi
```

- This runs a GPU job named “test_gpu_job”, with 2 CPUs and 1 MIG. Wall time is 20 minutes.

Check sample job scripts in `/apps/testjobs/workshop`

Limitation of GPU Jobs

- You cannot run your job using multiple MIG instances.
- For example, `--gres=gpu:a100_10g:2` will allocate two instances of the 10G MIG, but it will either raise an error, or some jobs may assume it as a single MIG, even if multiple instances are requested.

Manage Jobs – Submit via CLI

Submit a job script

- **$ sbatch my_script**
- Submitted batch job 1234

Listing jobs

- For current user in Pending, Running, Suspended states:
  - **$ squeue -u $LOGNAME**

```
JOBID PARTITION   NAME   USER ST TIME  NODES NODELIST(REASON)
1234   general uname.sh  test PD 0:00    2     (Priority)
```

For a more detailed query on active job:

- **$ scontrol show jobid=1234**

```
JobId=2 JobName=simple.job
UserId=test(1001) GroupId=test(1001) MCS_label=N/A
Priority=4294901759 Nice=0 Account=(null) QOS=standard
JobState=COMPLETED Reason=None Dependency=(null)
…
```

Canceling jobs

- **$ scancel 1234 -** Cancel job ID 1234
- **scancel --me** - Cancel all your jobs

Show information about an active or completed job

- **$ slurm_jobid 1234**

Common inquiries

checkq	• squeue but more details
slurm_jobid	• Show information about a running or queued job
sq	• Display pending job/queue info in a helpful way, You can also check the last job details with `sq`
squeue --start	• Jobs will be listed in order expected start time • Times are only estimates and subject to change
quota_info	• Show space and SU quotas for self or others

Some Common Problems

After using sbatch, the job disappears in 30 seconds and there's no result output.

- Check the details with slurm_jobid [JOBID], use --err and --out
- Use `sq` if you are unsure about the job id.

Invalid account or account/partition combination specified.

- Check `--account`
- Use `quota_info $LOGNAME`

```
JOBID PARTITION   NAME   USER ST TIME  NODES NODELIST(REASON)
1234    course uname.sh  xiss PD 0:00   2 (ReqNodeNotAvail,
Reserved for maintenance)
```

- Jobs that do not end before the maintenance window begins will be held until the maintenance is complete

Some Common Problems(contd)

```
JOBID PARTITION   NAME   USER ST TIME  NODES
NODELIST(REASON)
1234    course uname.sh  xiss PD 0:00      2
(AssocGrpBillingMinutes)
```

- You have reached the limit of SU.

Error message: cggroup out of memory handler

- Check the memory requirement
- You probably need to increase memory on `--mem` or the number of cpus from `--mem-per-cpu`
- If nothing works, then the problem is likely due to incorrect setup of the problem.

Using Applications with GUI on OnDemand

Login to ondemand.njit.edu

Go to “Interactive Apps” and select the application from the list.

If you don’t find the app, select Linux Desktop

Once are you connected, select “Terminal Emulator” from “Applications” option from top left

Interactive Batch Jobs

Interactive, but handled through batch system

Resource limits same as standard batch limits

Useful for tasks forbidden on login nodes

Debug parallel programs

Quickly test your code

May not be practical when system load is high

Long wait, same as standard batch job

To submit an interactive batch job (example)

- `interactive -a ACCOUNT -q QOS -p PARTITION -j JOB_TYPE`
- `interactive -h` for details

Reminder

- Wulver will be temporarily out of service for maintenance once a month, specifically on the 2nd Tuesday, to perform updates, repairs, and upgrades.
- During the maintenance period, the logins will be disabled
- Jobs that do not end before the maintenance window begins will be held until the maintenance is complete
  - Reduce the walltime in the job script to run the job

- Date: Every Monday and Wednesday
- Time: 2:00–4:00 p.m.
- Location: GITC 5320N
- Meet with our student consultants and ask any questions you have about using HPC resources.
- There's no need to create a ticket in advance.

Resources to get your questions answered

Getting Started: Access to Wulver

List of Software: Wulver Software

HOW TOs: Conda Documentation

Installing Python packages via Conda

Request Software: HPC Software Installation

Access to OnDemand Open OnDemand

MIG Information MIG

Contact: Please visit HPC Contact

Open a ticket: email to hpc@njit.edu

Consult with Research Computing Facilitator: Facilitator Calendar Appointment

System updates

- Read Message of the Day on login
- Visit NJIT HPC News

NJIT

hpc@njit.edu

hpc.njit.edu