Ohio Supercomputer Center

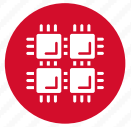
An OH·TECH Consortium Member

Workshop Set up

- Workshop project – set up account at my.osc.edu
  - **If you already have an OSC account, sign in to my.osc.edu**
    - **Go to Project**
      - **Project Access Request**

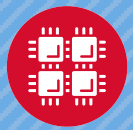
PROJECT CODE = PZS0724

- Slides are on event page: osc.edu/events
- Workshop website:
  - https://khill42.github.io/OSC_IntroHPC/



Computing Services to Accelerate Research and Innovation

An introduction to OSC services, hardware, and environment



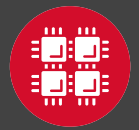
Kate Cahill

Education & Training Specialist

“OSC is here to empower your research.”

About OSC

- Founded in 1987
- Statewide resource for all universities in Ohio
  - high performance computing services
  - computational science expertise
  - “ … propel Ohio's research universities and private industry to the forefront of computational based research.”
- Funded through the Ohio Department of Higher Education
- Reports to the Chancellor
- Located on OSU’s west campus
- Fiscal agent is OSU



The OH-TECH Consortium

Ohio Supercomputer Center provides high performance computing, software, storage and support services for Ohio's scientists, faculty, students, businesses and their research partners.

OARnet connects Ohio's universities, colleges, K-12, health care and state and local governments to its high-speed fiber optic network backbone. OARnet services include co-location, support desk, federated identity and virtualization.

OhioLINK serves nearly 600,000 higher education students and faculty by providing a statewide system for sharing 50 million books and library materials, while aggregating costs among its 90 member institutions.

Service Catalog

Cluster Computing

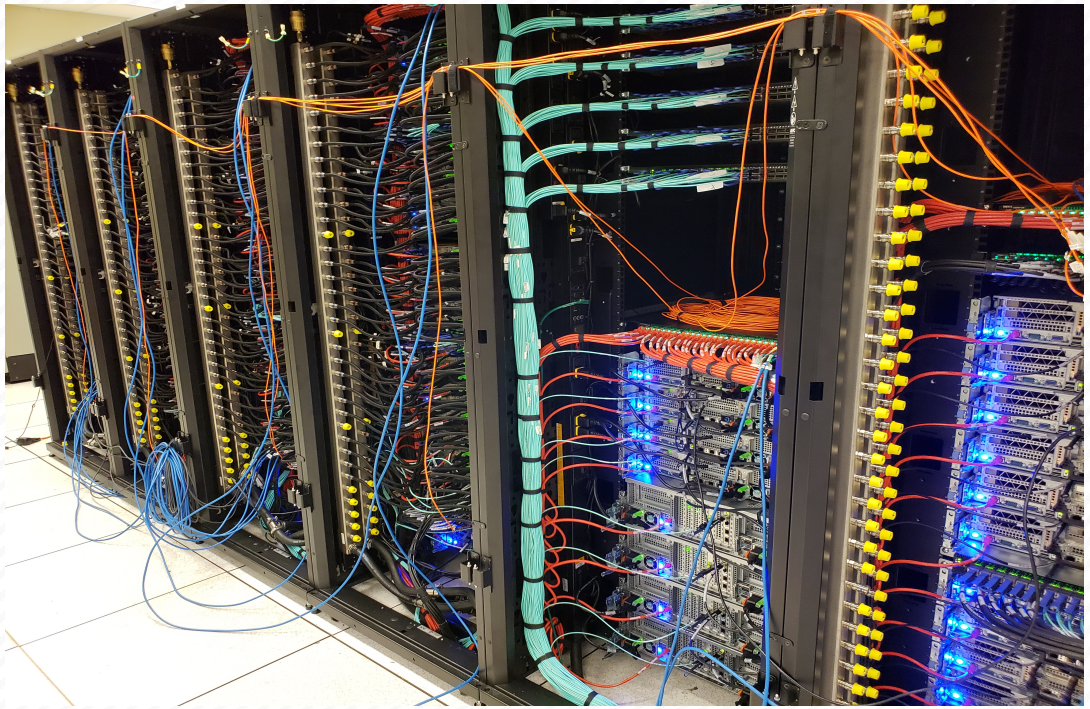
A fully scalable center with mid-range machines to match those found at National Science Foundation centers and other national labs.

Research Data Storage

High-performance, large capacity data storage spaces along with others that are perfect for a wide variety of research data.

Education

High performance computing and networking resources come together to create an exciting and innovative teaching and research environment.



Web Software Development

Our expert web development team helps you create custom web interfaces to simplify the use of powerful HPC resources.

Scientific Software Development

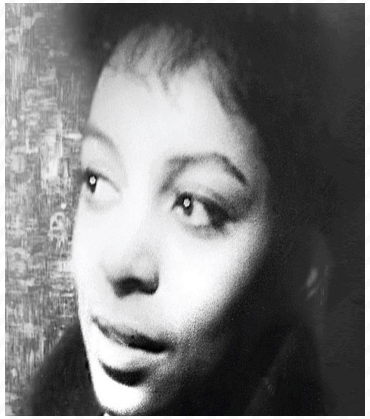
Deep expertise in developing and deploying software that runs efficiently and correctly on large scale cluster computing platforms.

Hardware Overview

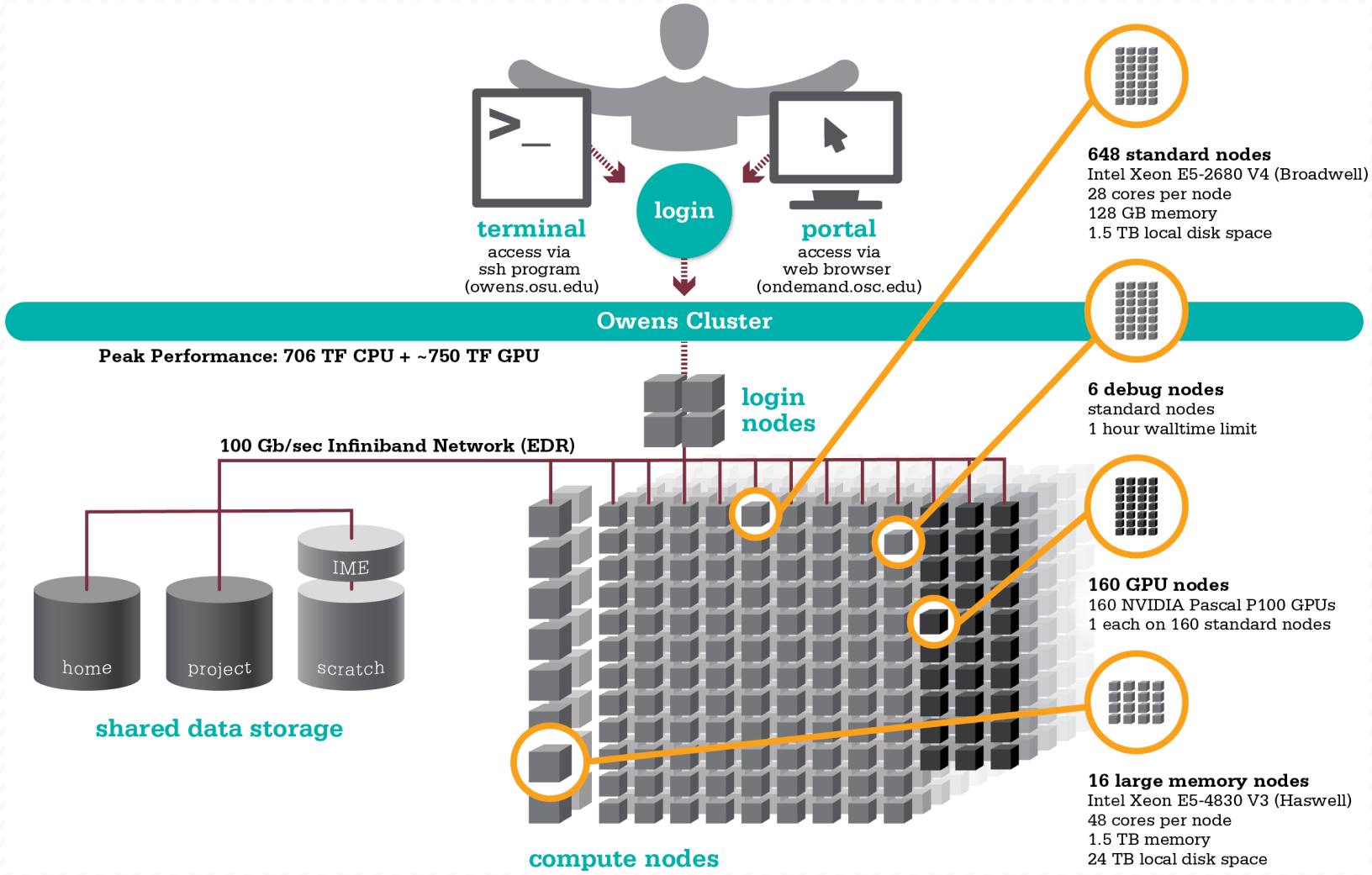
“To err is human, but to really foul things up you need a computer.” – Paul Ehrlich

System Configurations

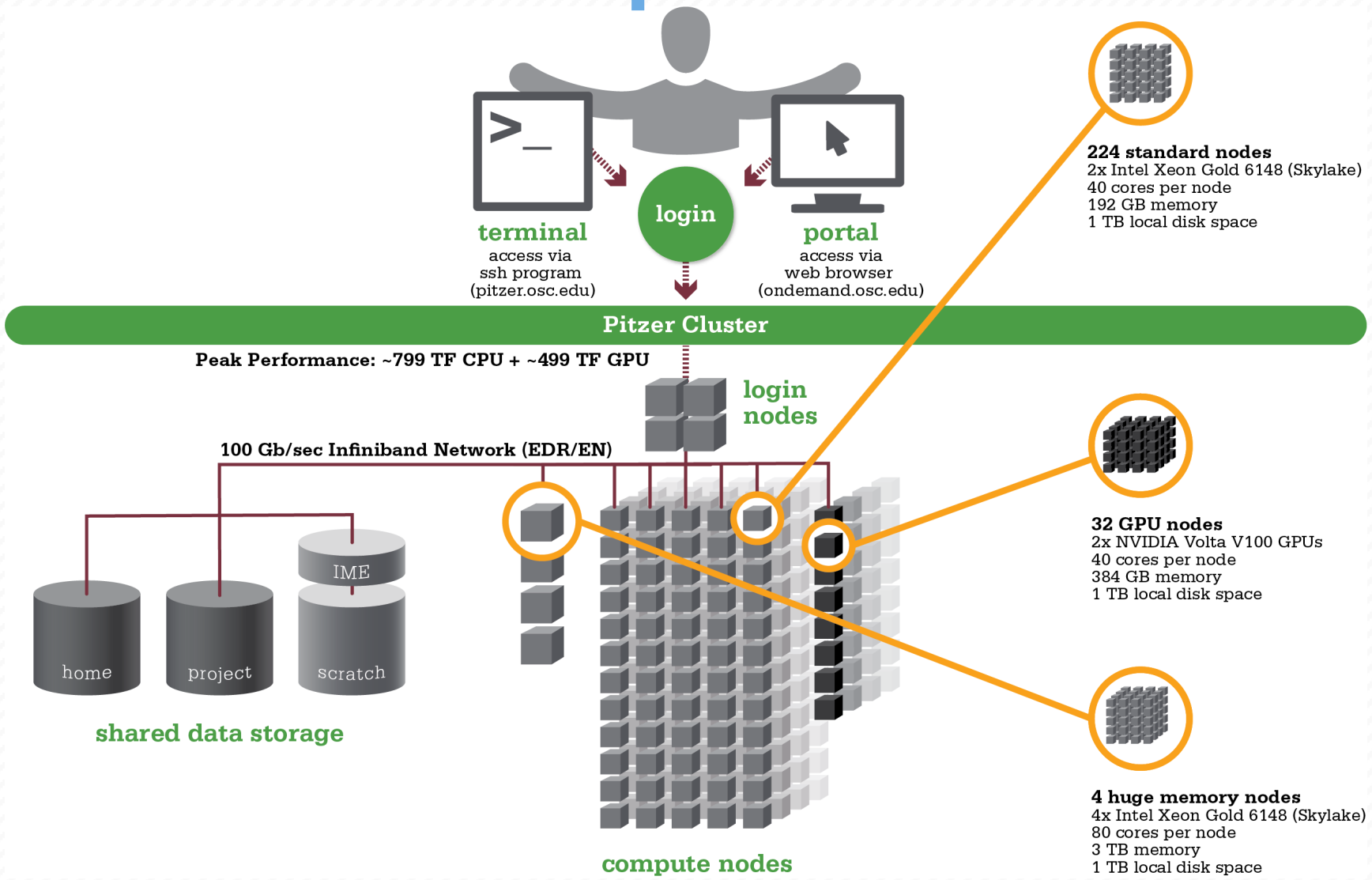
	Pitzer (2018)	Owens (2016)	Ruby (2014)
Theoretical Performance	~1300 TF	~1600 TF	~144 TF
# Nodes	260	824	240
# CPU Cores	10,560	23,392	4,800
Total Memory	~70.6 TB	~120 TB	~15.3 TB
Memory per Core	>5 GB	>5 GB	3.2 GB
Interconnect	EDR IB	EDR IB	FDR/EN IB



Owens Cluster Specifications

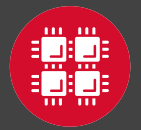


Pitzer Cluster Specifications



Login Nodes – Usage

- Purpose
  - Submit jobs to batch system
  - Edit files
  - Manage your files
  - Interactive work – small scale
- Limits
  - 20 minutes CPU time
  - 1GB memory
- **Use the batch system for serious computing!**



Resources to get your questions answered

FAQs: https://www.osc.edu/resources/getting_started/supercomputing_faq

HOW TOs: https://www.osc.edu/resources/getting_started/howto

New User Guide: https://www.osc.edu/resources/getting_started/new_user_resource_guide

Updated presentations: https://www.osc.edu/~kcahill/NewUser

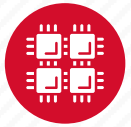
Office Hours:

go.osu.edu/rc-osc Alternate Tuesdays 1-3 p.m. at Research Commons

Walk in: Wed & Fri. 1-2:30 p.m. at Pomerene Hall

System updates

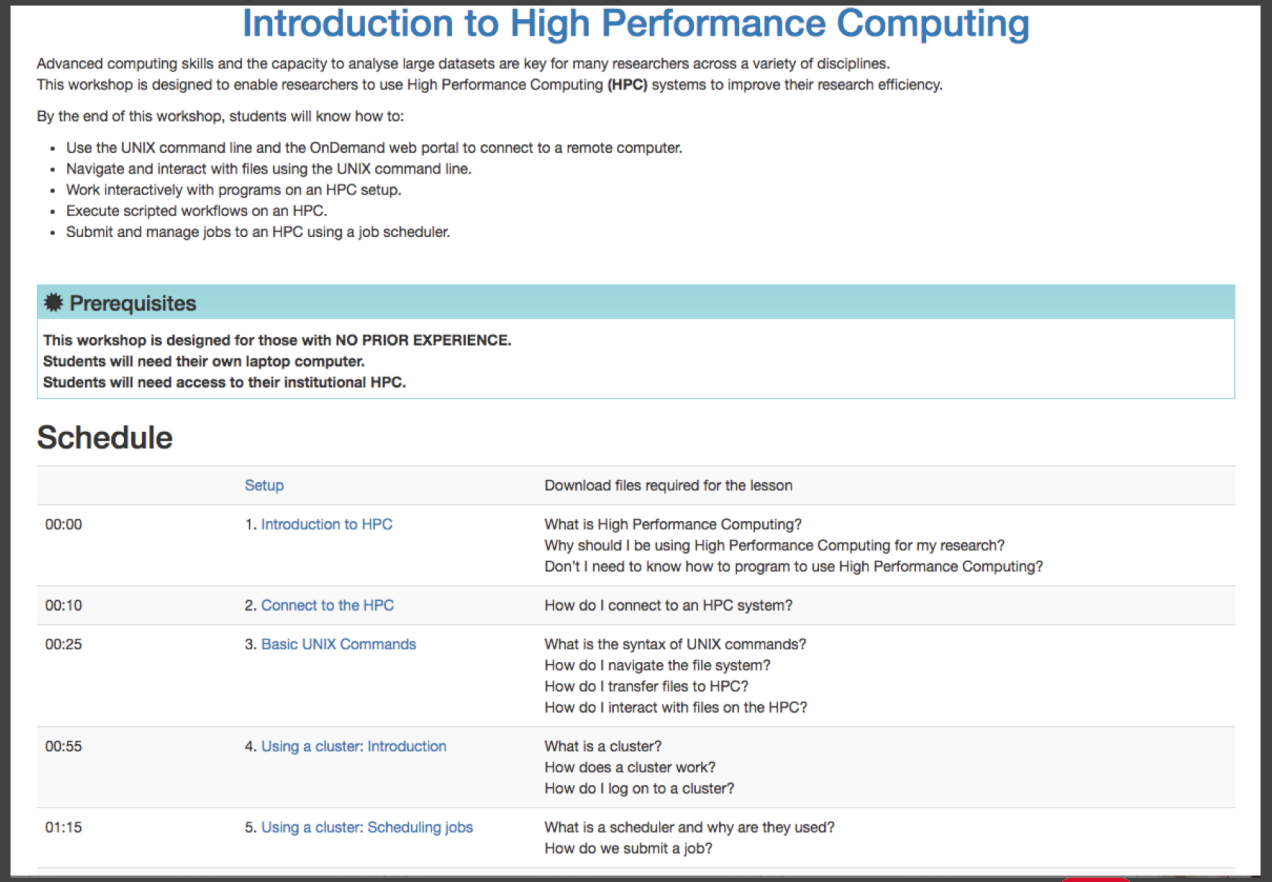
- Read Message of the Day on login
- Follow @HPCNotices on Twitter



Intro to OSC tutorial

https://khill42.github.io/OSC_IntroHPC/

- Highlights OnDemand for OSC access
- Always available training
- Continually being updated and improved
- Based on HPC/Data carpentry materials
- OSC Workshop Oct. 10



Introduction to High Performance Computing

Advanced computing skills and the capacity to analyse large datasets are key for many researchers across a variety of disciplines.
This workshop is designed to enable researchers to use High Performance Computing **(HPC)** systems to improve their research efficiency.

By the end of this workshop, students will know how to:

- Use the UNIX command line and the OnDemand web portal to connect to a remote computer.
- Navigate and interact with files using the UNIX command line.
- Work interactively with programs on an HPC setup.
- Execute scripted workflows on an HPC.
- Submit and manage jobs to an HPC using a job scheduler.

Prerequisites

This workshop is designed for those with NO PRIOR EXPERIENCE.
Students will need their own laptop computer.
Students will need access to their institutional HPC.

Schedule

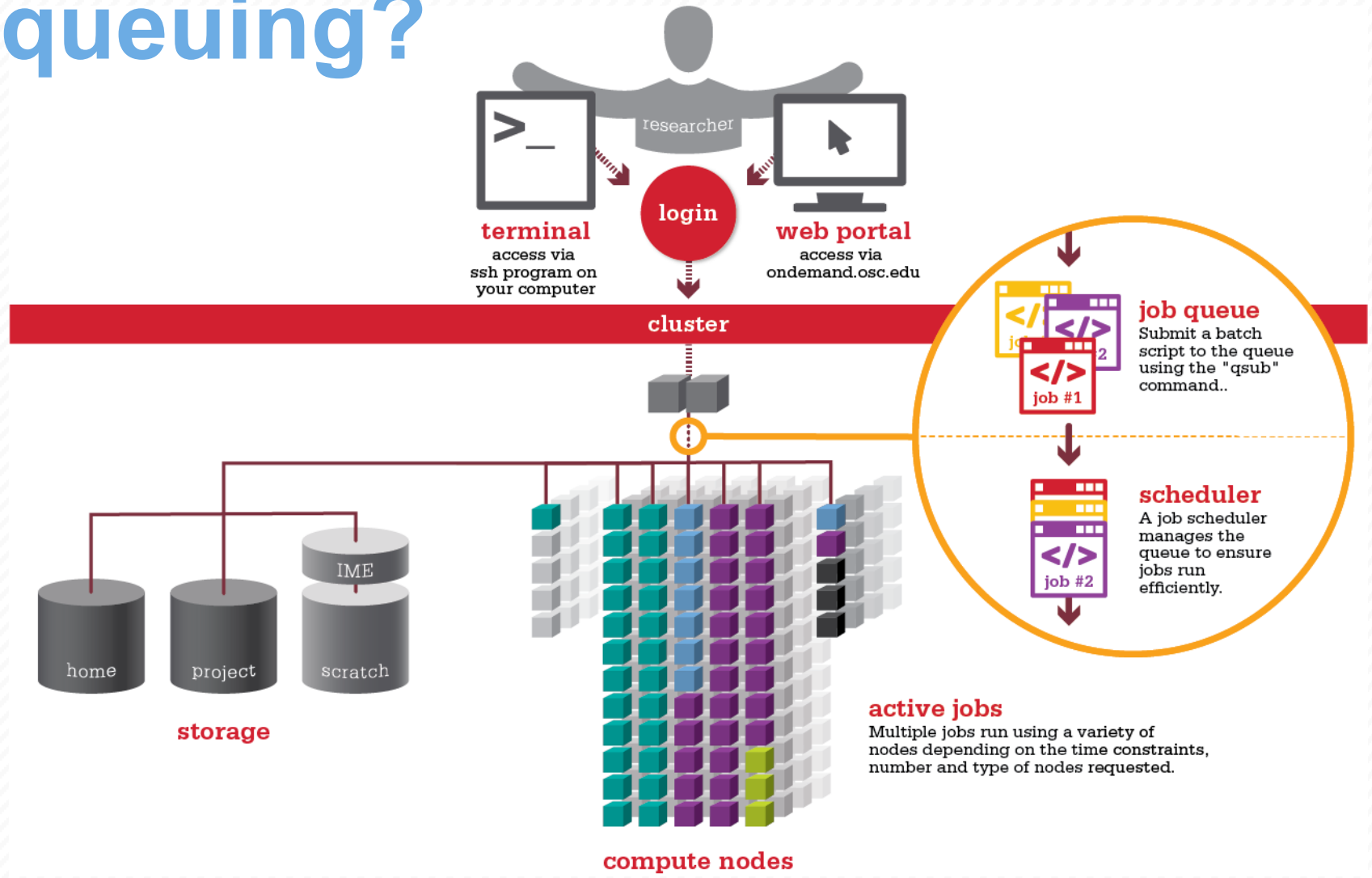
	Setup	Download files required for the lesson
00:00	1. Introduction to HPC	What is High Performance Computing? Why should I be using High Performance Computing for my research? Don't I need to know how to program to use High Performance Computing?
00:10	2. Connect to the HPC	How do I connect to an HPC system?
00:25	3. Basic UNIX Commands	What is the syntax of UNIX commands? How do I navigate the file system? How do I transfer files to HPC? How do I interact with files on the HPC?
00:55	4. Using a cluster: Introduction	What is a cluster? How does a cluster work? How do I log on to a cluster?
01:15	5. Using a cluster: Scheduling jobs	What is a scheduler and why are they used? How do we submit a job?


Questions?

Batch Processing

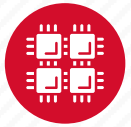
“There’s an old story about the person who wished his computer were as easy to use as his telephone. That wish has come true, since I no longer know how to use my telephone.” – Bjarne Stroustrup

Why do supercomputers use queuing?



Idea Behind Batch Processing

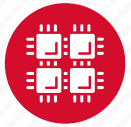
- Whatever you would normally type at the command prompt goes into your batch script
- Output that would normally go to the screen goes into a log file (or files)
- The system runs your job when resources become available
- Very efficient in terms of resource utilization



Batch System at OSC

- Compute nodes are allocated through the batch system
  - PBS – Portable Batch System
  - Torque – resource manager
  - Moab – scheduler
- Documentation at

  www.osc.edu/supercomputing/batch-processing-at-osc

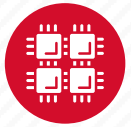


Steps for Running a Job on the Compute Nodes

1. Create a batch script for a job
2. Submit the job
3. Job gets queued
4. Job runs when resources become available
5. Get your results when the job finishes

Specifying Resources in a Job Script

- Nodes and cores (processors) per node, GPUs
- Memory (optional)
- Walltime
  - Overestimate slightly – job will be deleted if it hits limit
  - Shorter job may start sooner due to backfill
- Project #
- Software licenses
  - See specific software page on OSC website



Sample Batch Script

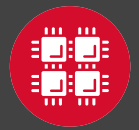
```
#PBS -N serial_fluent
#PBS -l walltime=1:00:00
#PBS -l nodes=1:ppn=28:gpus=1
#PBS -A PAS####
#PBS -j oe
#PBS -l software=fluent+1
# Set up the FLUENT environment
module load fluent
# Move to directory job was submitted from
cd $PBS_O_WORKDIR
# Copy input files to compute node
cp run.input $TMPDIR
cd $TMPDIR
# Run fluent and copy results back to home
fluent 3d -g < run.input
cp 'results*' $PBS_O_WORKDIR
```

Job setup information for PBS

This is a comment

Commands to be run

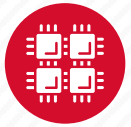
Put all this into a text file!



Submitting a Job and Checking Status

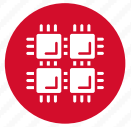
- Command to submit a job
  - **`qsub` *`script_file`***
- Response from PBS (example)
  - `123456.owens-batch.ten.osc.edu`
- Show status of batch jobs
  - **`qstat -a` *`jobid`***
  - **`qstat -u` *`username`***
  - **`qstat -f` *`jobid`***
- Delete a batch job
  - **`qdel` *`jobid`***

List of Batch commands on osc.edu



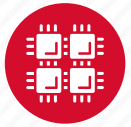
Scheduling Policies and Limits

- Walltime limit
  - 168 hours for serial jobs (single node)
  - 96 hours for parallel jobs (multiple nodes)
- Per-user limits
  - 128 concurrently running jobs
  - 2040 processor cores in use
  - 1000 jobs in the batch system, running or queued
- Per-group limits
  - 192 concurrently running jobs
  - 2040 processor cores in use



Waiting for Your Job To Run

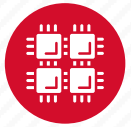
- Queue wait time depends on many factors
  - System load
  - Resources requested
    - nodes, cores, large memory, gpus, software licenses
  - Fair share limits (if load is high)
    - reduced priority for users or groups using a lot of resources



Job Output

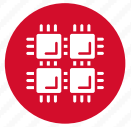
- Screen output ends up in file ***job_name.ojobid***
  - Copied to your working directory when job ends
  - Example: **testjob.o1234567**
- To see screen output while job is running
  - **qpeek *jobid***
  - Example: **qpeek 1234567**

```
----------------------
Resources requested:
nodes=2:ppn=28
----------------------
Resources used:
cput=125:18:32
walltime=02:14:32
mem=34.824GB
vmem=77.969GB
----------------------
Resource units charged (estimate):
12.556 RUs
----------------------
```



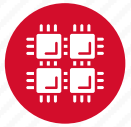
Interactive Batch Jobs

- Interactive, but handled through batch system
  - Resource limits same as standard batch limits
- Useful for tasks forbidden on login nodes
  - Debug parallel programs
  - Run a GUI program that's too large for login node
- May not be practical when system load is high
  - Long wait, same as standard batch job
- To submit an interactive batch job (example)
  - `qsub -I -X -l nodes=2:ppn=28 -l walltime=1:00:00 -m abe`



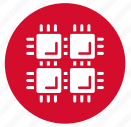
Batch Queues

- The three clusters have separate batch systems
  - Submit job and check status on the same cluster
- Debug reservation
  - A few nodes on each system are reserved for short jobs (≤ 1 hour)
  - Special flag required: `qsub -q debug job_script.sh`



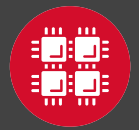
Parallel Computing

- Each processor is fast, but real speed comes from using multiple processors
- Multithreading
  - Use multiple cores on a single node
  - Shared memory
- Message passing (MPI)
  - Use one or multiple nodes
  - Distributed memory



To Take Advantage of Parallel Computing

- Program must be written to take advantage of multiple cores and/or multiple nodes
- Many commercial applications have multithreaded or parallel versions
- Must use `mpiexec` for multiple nodes
- **Can’t just request more nodes or cores and expect your job to run faster**



OH·TECH

Ohio Technology Consortium

A Division of the **Ohio Department of Higher Education**

info@osc.edu

osc.edu

twitter.com/osc

oh-tech.org/blog

facebook.com/ohiosupercomputercenter

linkedin.com/company/ohio-supercomputer-center