Ohio Supercomputer Center

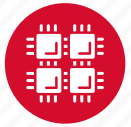
An OH·TECH Consortium Member

Workshop Set up

- Workshop website:
  - https://khill42.github.io/OSC_IntroHPC/

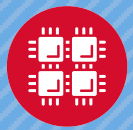
- Workshop project – set up account at my.osc.edu
  - PZS0724
  - **Nq7sRoNrWnFuLtBm**

  - **If you already have an OSC account, sign in to my.osc.edu**
    - **Go to Project**
      - **Project Access Request**



Computing Services to Accelerate Research and Innovation

An introduction to OSC services, hardware, and environment



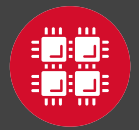
Kate Cahill

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

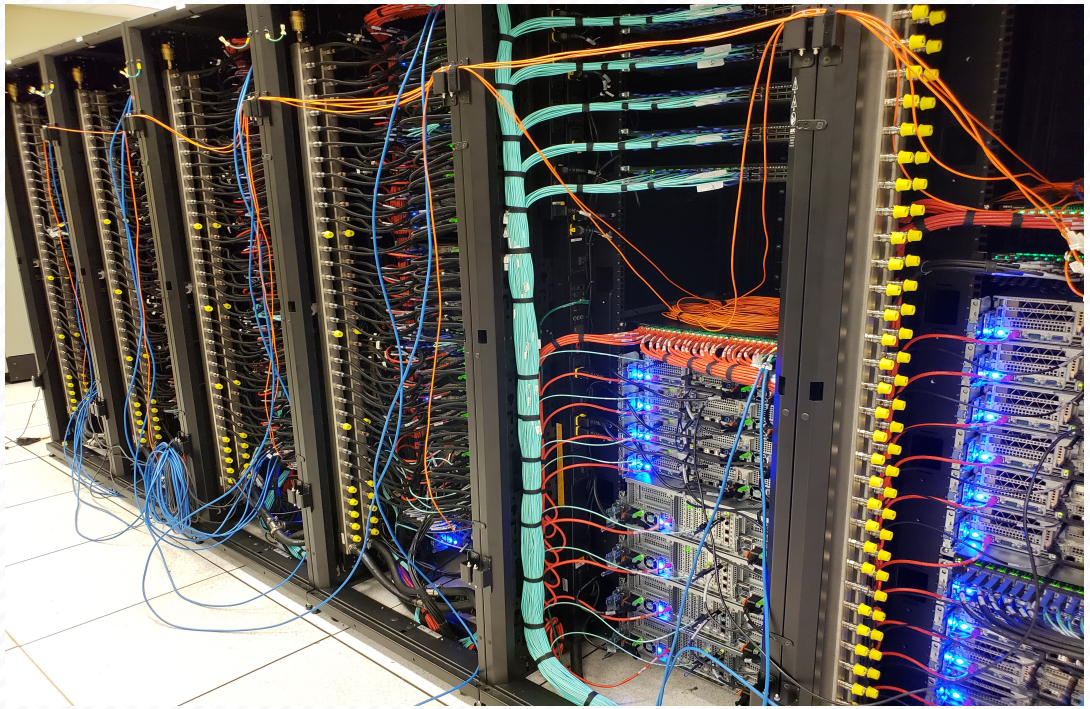
A fully scalable center with mid-range machines to match those found at National Science Foundation centers and other national labs.

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

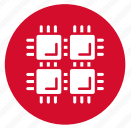
Our expert web development team helps you create custom web interfaces to simplify the use of powerful HPC resources.

Scientific Software Development

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

HPC Example Projects and Concepts

“The difference between us and a computer is that, the computer is blindingly stupid, but it is capable of being stupid many, many million times a second.” – Douglas Adams



Why would HPC be necessary for your work?

- Your simulations or analyses take too long on your personal computer
- The size of your data is too large to be contained (storage) or accessed (memory) on your computer
- You would like to free up your own system to do other tasks
- You need particular software for your work

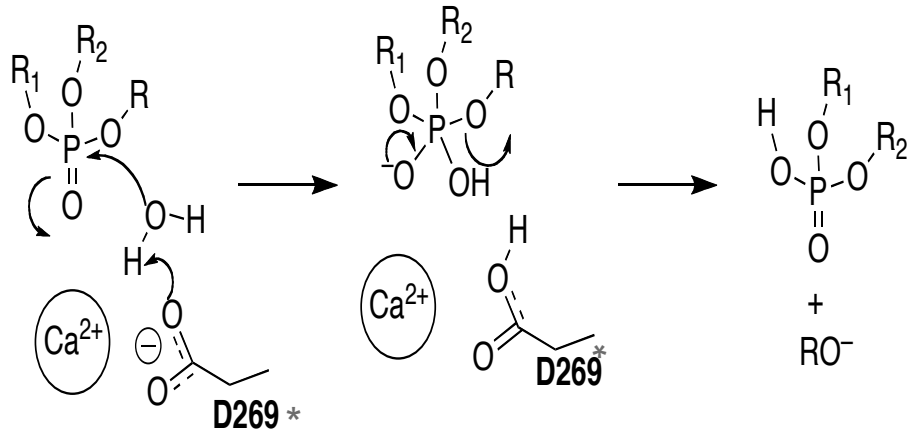
Mapping

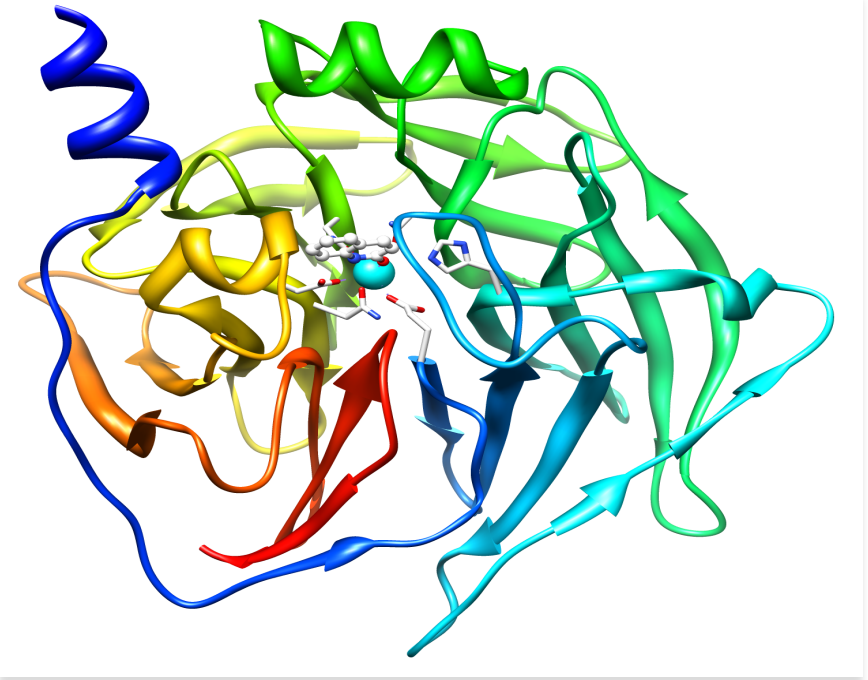
Researchers who normally use OSC systems to enhance satellite images of glaciers turned their technology to disaster relief assistance following Nepal's April 2015 earthquake

PI: Ian Howat, Ohio State University

Source: NSF Office of Polar Programs

Treating Nerve Agent Exposure





Treating Nerve Agent Exposure

With the power of OSC computing services, a team studies how to capture and destroy organophosphorus nerve agents using modified proteins.

PI: Christopher Hadad, Ohio State University

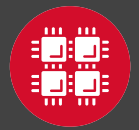
What is the difference between your laptop and a supercomputer?

X 10000 =

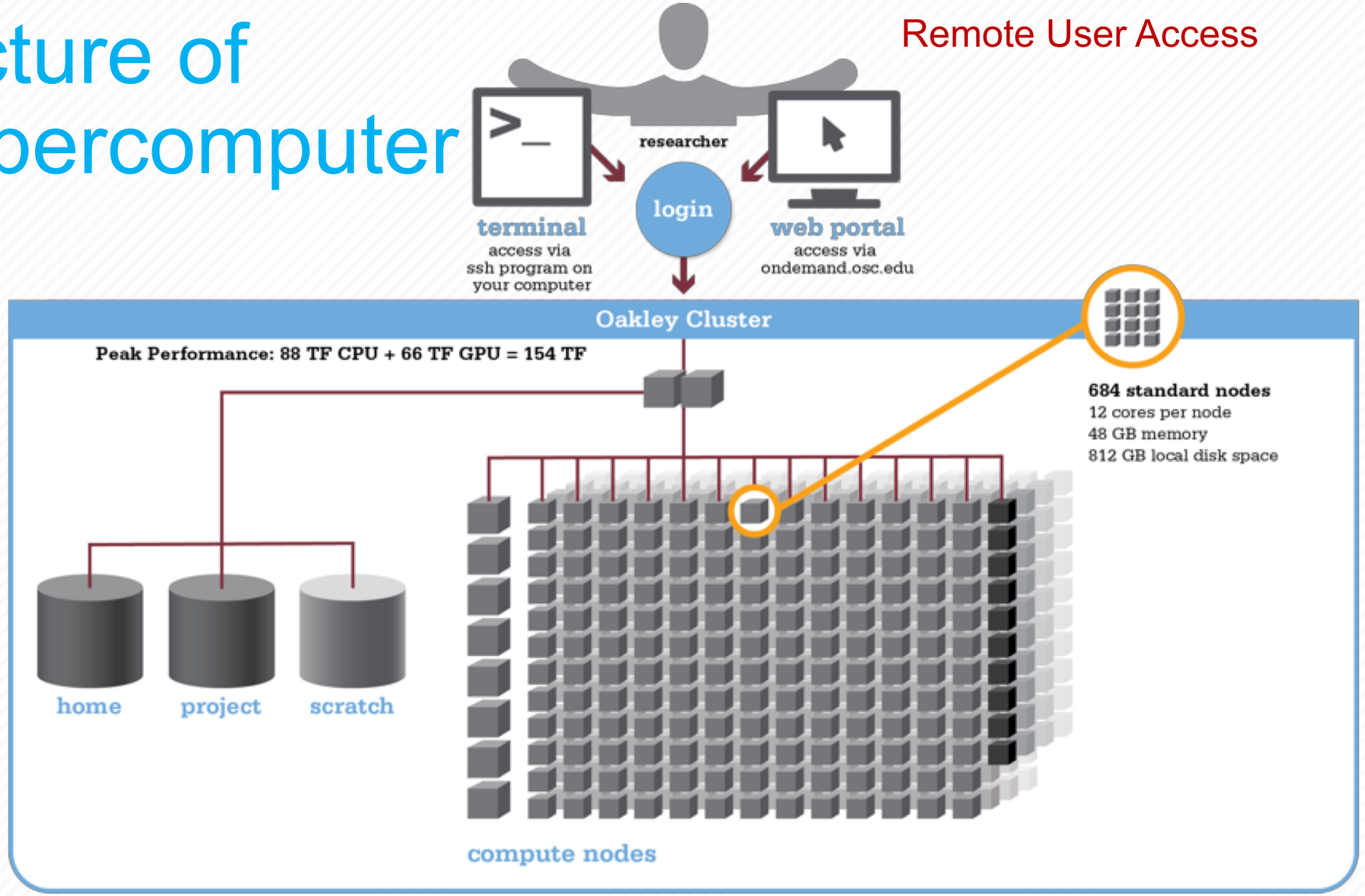
Remote Connect

HPC Terminology

- Cluster
  - A group of computers (nodes) connected by a high-speed network, forming a supercomputer
- Node
  - Equivalent to a high-end workstation, part of a cluster
- Core
  - A processor (CPU), multiple cores per processor chip
- FLOPS
  - “FLoating-point Operations (calculations) Per Second”
- GPU (Graphical Processing Unit)
  - A separate multi-core processor that can handle many small calculations

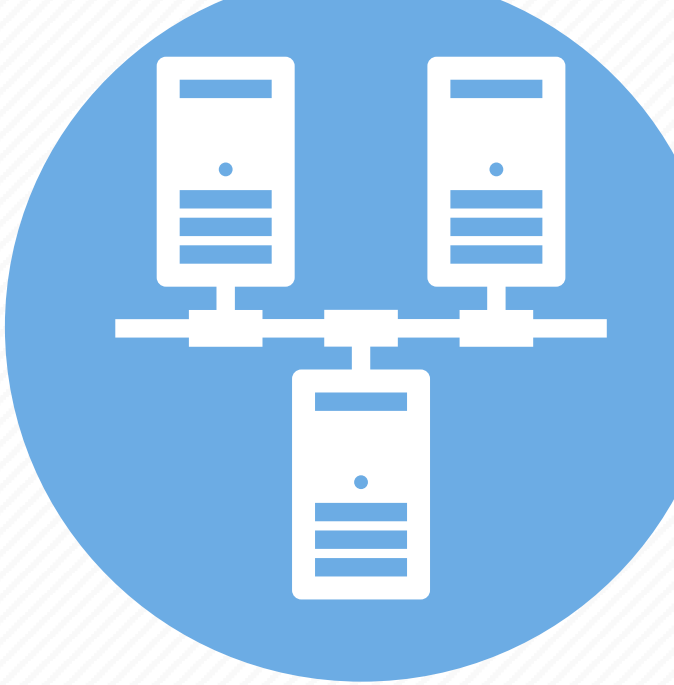


Structure of a Supercomputer



Remote User Access

Memory



- Holds data that is being calculated on, as well as computational instructions
- *Shared memory* is local to one node and several process

threads can share the same data addresses.

- *Distributed memory* is on multiple nodes and each

process normally has its own copy or part of the data.

Storage

Different types of “disk” for different needs

- Local disk in the node, often SSD
- Shared scratch

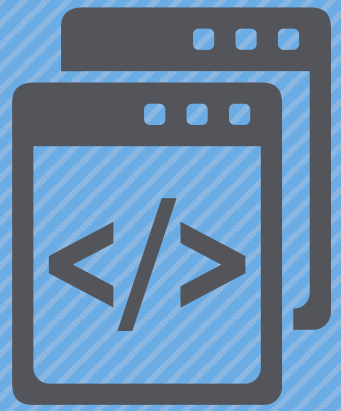
Parallel filesystems, eg Lustre or GPFS

Traditionally tuned for high bandwidth, not high IOPS

May have a “burst buffer” layer in front of it

Short-term storage only!!

- Longer-term or archive

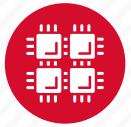


Getting Started at OSC

“If you were plowing a field, which would you rather use? Two strong oxen or 1024 chickens?” - Seymour Cray

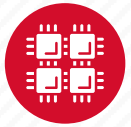
Who can get an OSC project?

- Academic project
  - Principal investigator (PI) must be a full-time faculty member or research scientist at an Ohio academic institution
  - PI may authorize accounts for students, post-docs, collaborators, etc.
  - Classroom projects are also available
  - No cost to Ohio academic users
- Commercial projects
  - Commercial organizations may purchase time on OSC systems



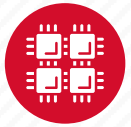
Accounts and Projects at OSC

- Project
  - Headed by a PI
  - May include other users
  - Basis for accounting at OSC
  - Submit proposal for computing resources for a project
- Account
  - Username and password to access HPC systems
  - Each account used by one person (please!)
  - If you work on multiple projects, you will have one account that can access all of them



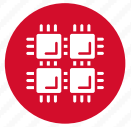
Allocations and Charges

- Charges are in terms of resource units
- Resource units
  - OWENS & PITZER 1 resource unit (RU) = 10 CPU hours
  - RUBY 1 resource unit (RU) = 20 CPU hours
  - CPU hour = walltime x (total # of cores requested)
- Project receives an allocation of RUs
- Jobs are charged to a project



Fee structure

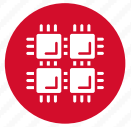
- The first 10,000 RUs on all academic projects are subsidized
- Usage above 10,000 RUs will be billed to the PI's institution at $0.075/RU
- PIs should contact their Office of Research for details on local billing



Requesting a New Project-

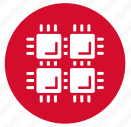
https://www.osc.edu/supercomputing/support/account

- Startup grant
  - One per PI per lifetime
  - Provide contact info, institution, department
  - 5000 RUs
- Additional allocations for a project
  - Submit a proposal for more RUs
    - Standard: 10,000
    - Major: 30,000
    - Discovery: >30,000
  - Peer-reviewed
  - Grants awarded by Statewide Users Group (SUG)
- Classroom accounts
- Request at my.osc.edu



My.osc.edu

- Site for managing your identity at OSC
- Update your email
- Change your password
- Recover access to your account
- Change your shell
- And a lot more in the future
  - Project reporting
  - Authorized user management
  - Requesting services (e.g. software access)



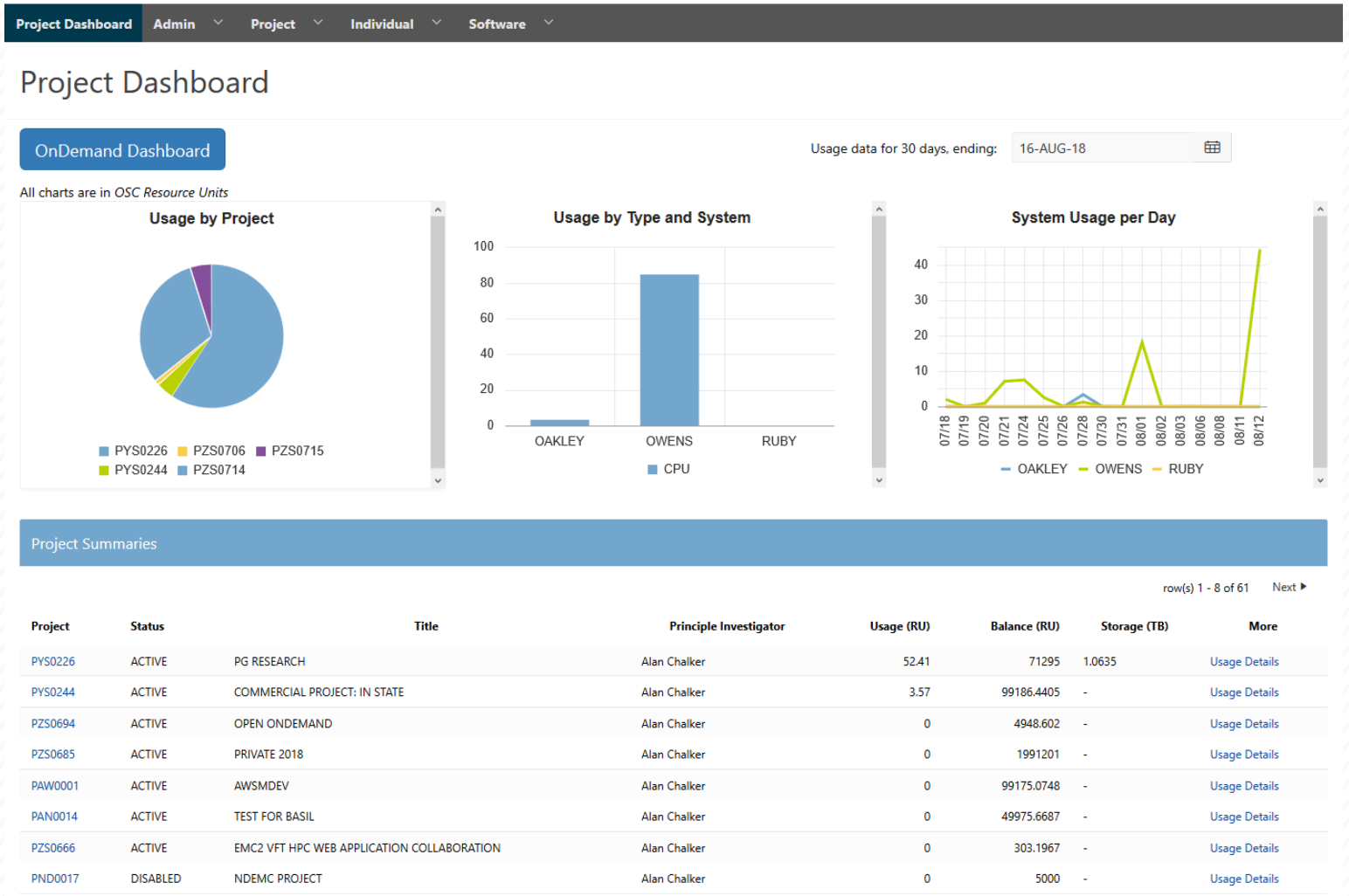
Client Portal– my.osc.edu

Features

- Create your account
- Update your email
- Change your password
- Recover access to your account
- Change your shell

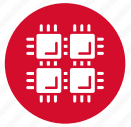
PI resources

- Project reporting
- Authorized user management
- Requesting services (e.g. software access)



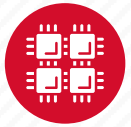
Your Contact Info

- Keep your contact information current
  - Use my.osc.edu to manage your account details.
- If your student continues to use OSC after graduation, make sure email address is updated
  - Acceptable if still collaborating with you
- May need to contact you about problems
- Will need to contact you about regular password changes
- You can opt out of routine notifications



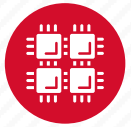
System Status

- Check system status on:
  - Message of the day (/etc/motd) – displayed at login
  - Twitter: @HPCnotices
  - Email for major outages or problems
- Scheduled downtimes
  - Quarterly maintenance – usually one day outage
  - Jobs held for up to two weeks prior
  - Last downtime was February 5



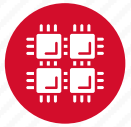
Statewide Users Group (SUG)

- The Statewide Users Group (SUG) is made up of OSC users
  - Provides program and policy advice to OSC
  - Meets twice a year
  - Headed by a chairperson elected yearly
- Standing committees
  - Allocations
  - Software and Activities
  - Hardware and Operations
- Get involved!
  - Next meeting will be April 18



Citing OSC

- Please cite OSC in your publications:
  - Details at www.osc.edu/citation
- These publications should be reported to OSC

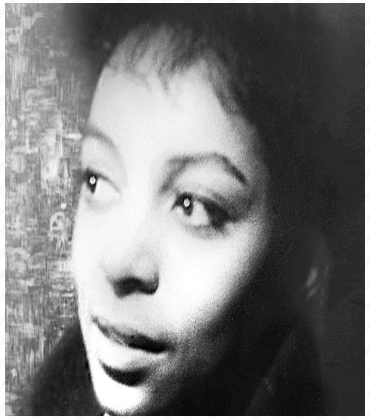


Hardware Overview

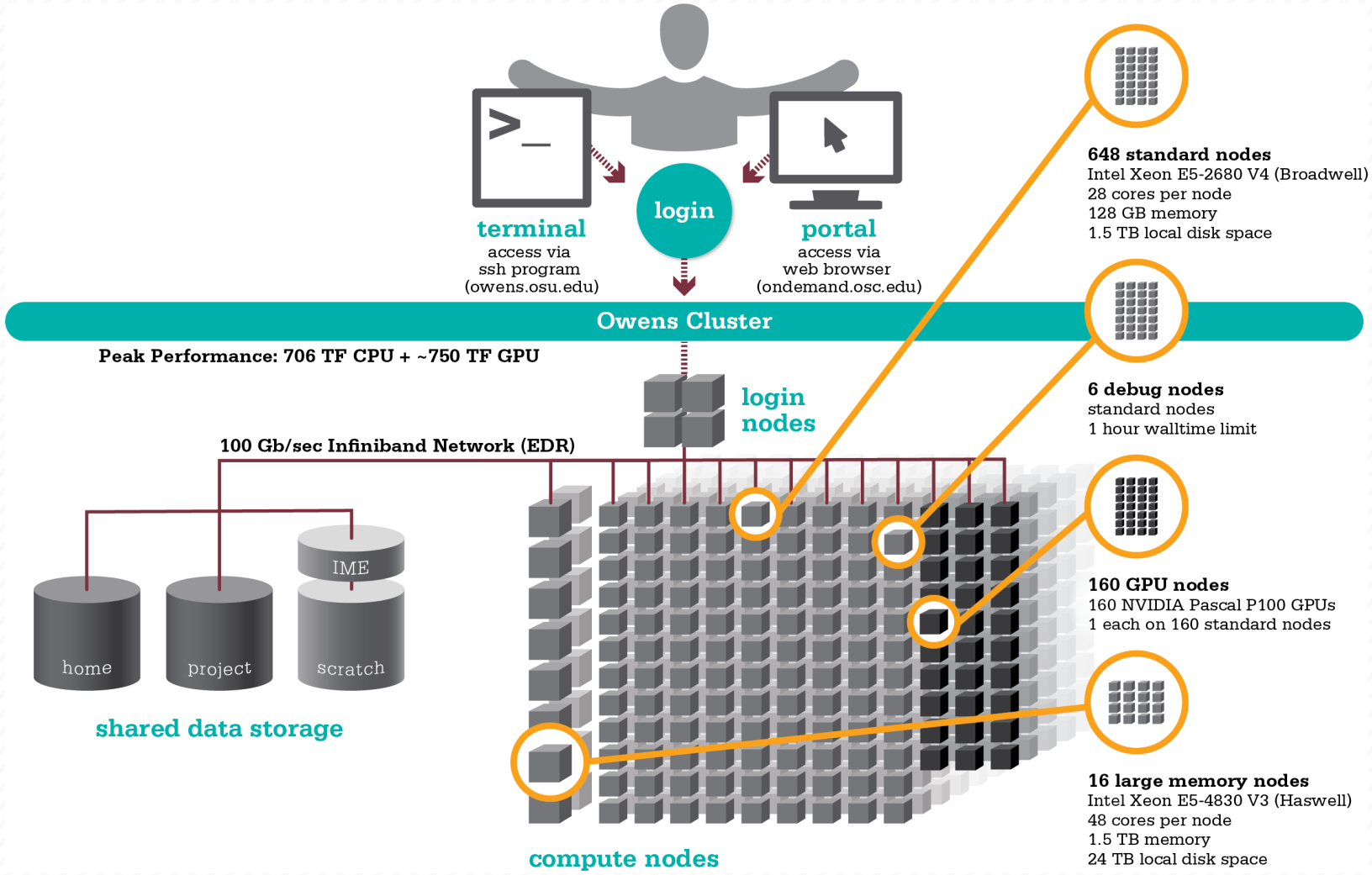
“To err is human, but to really foul things up you need a computer.” – Paul Ehrlich

System Configurations

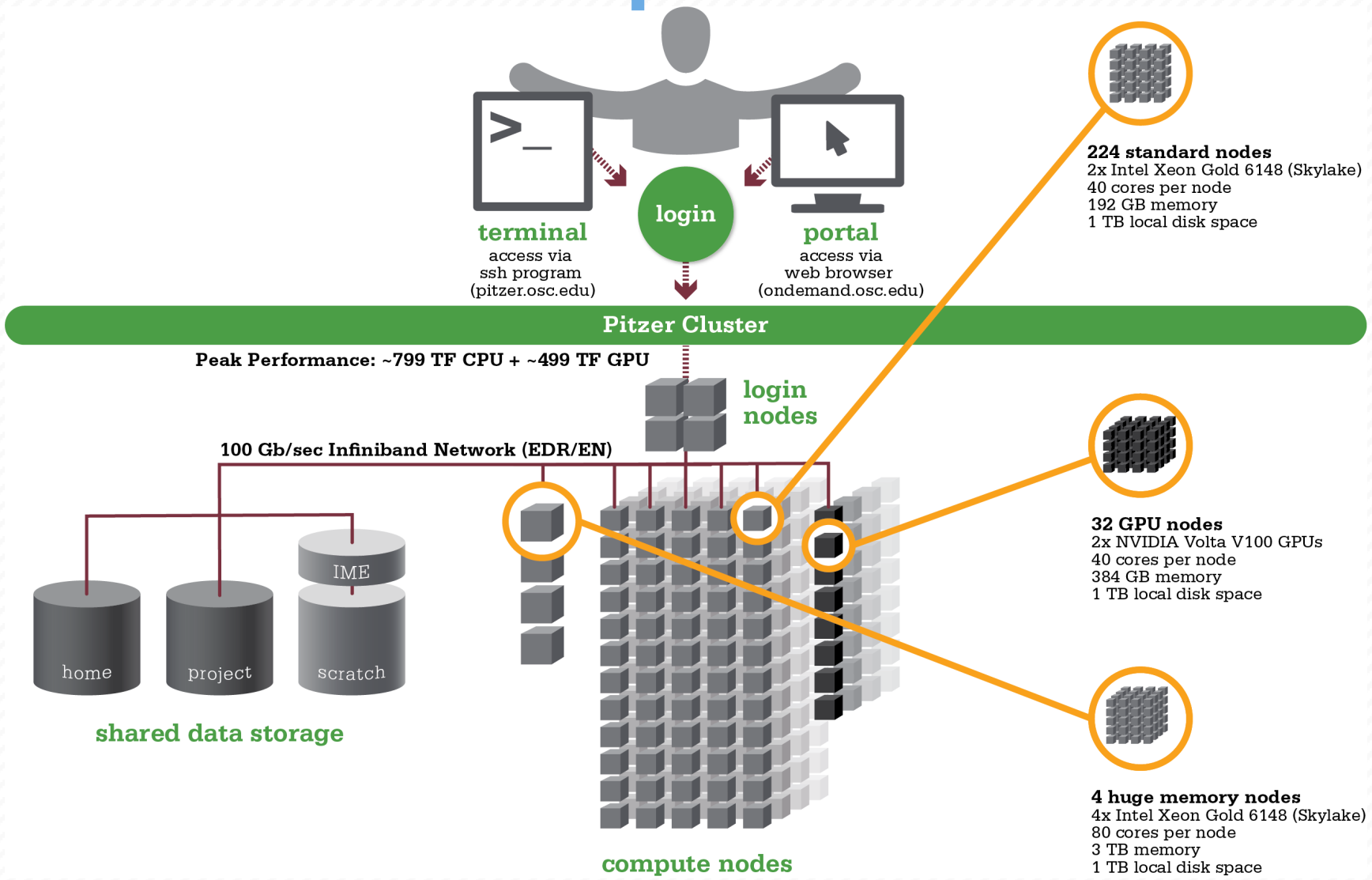
	Pitzer (2018)	Owens (2016)	Ruby (2014)
Theoretical Performance	~1300 TF	~1600 TF	~144 TF
# Nodes	260	824	240
# CPU Cores	10,560	23,392	4,800
Total Memory	~70.6 TB	~120 TB	~15.3 TB
Memory per Core	>5 GB	>5 GB	3.2 GB
Interconnect	EDR IB	EDR IB	FDR/EN IB



Owens Cluster Specifications

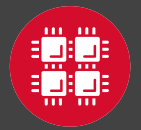


Pitzer Cluster Specifications



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

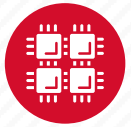
Office Hours:

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

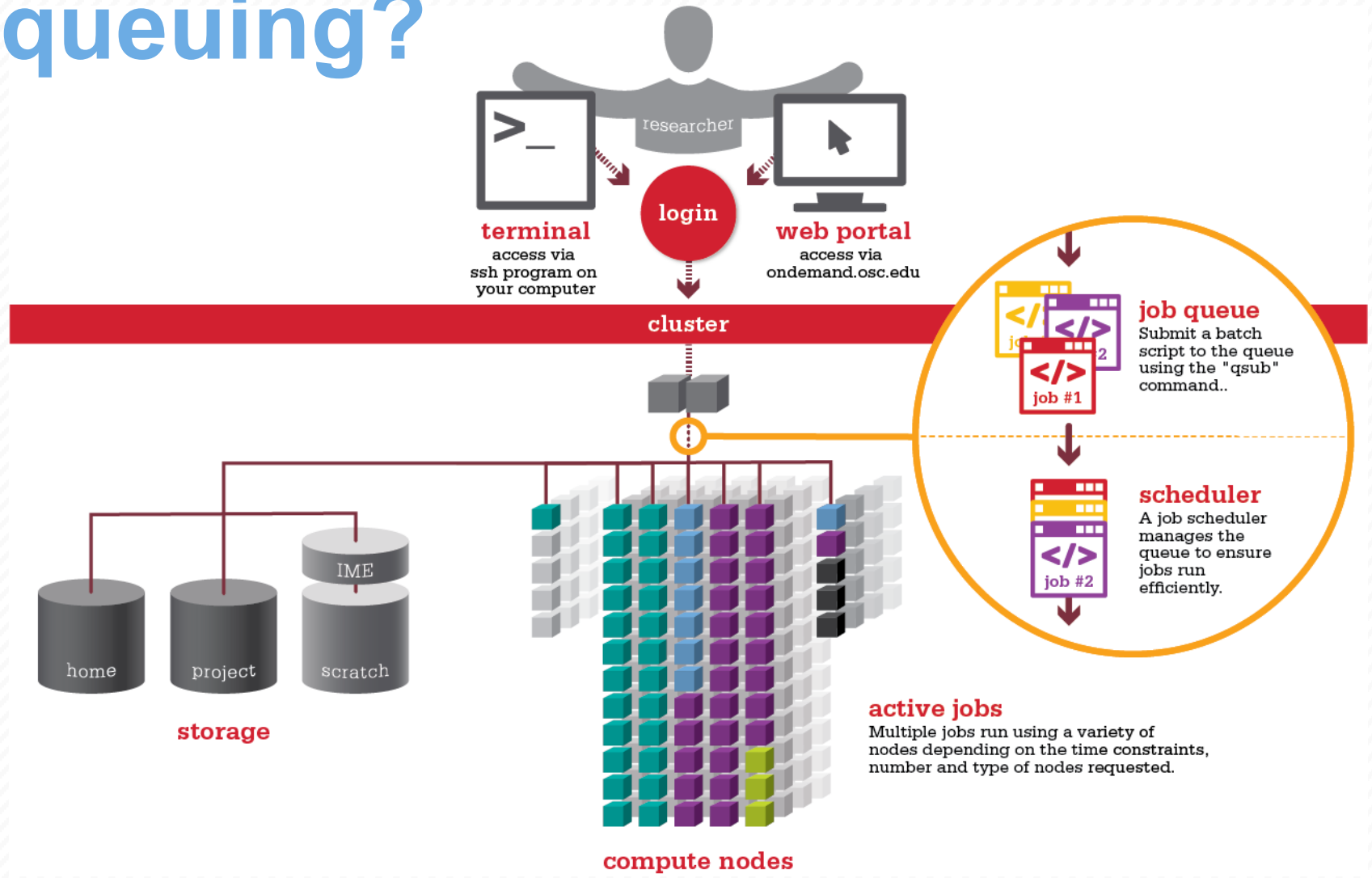



Questions?

Batch Processing

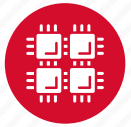
“There’s an old story about the person who wished his computer were as easy to use as his telephone. That wish has come true, since I no longer know how to use my telephone.” – Bjarne Stroustrup

Why do supercomputers use queuing?



Idea Behind Batch Processing

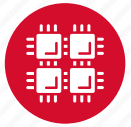
- Whatever you would normally type at the command prompt goes into your batch script
- Output that would normally go to the screen goes into a log file (or files)
- The system runs your job when resources become available
- Very efficient in terms of resource utilization



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

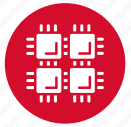


Steps for Running a Job on the Compute Nodes

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

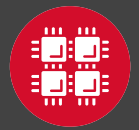
```
#PBS -N serial_fluent
#PBS -l walltime=1:00:00
#PBS -l nodes=1:ppn=28:gpus=1
#PBS -A PAS####
#PBS -j oe
#PBS -l software=fluent+1
# Set up the FLUENT environment
module load fluent
# Move to directory job was submitted from
cd $PBS_O_WORKDIR
# Copy input files to compute node
cp run.input $TMPDIR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

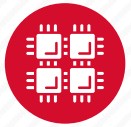
Put all this into a text file!



Submitting a Job and Checking Status

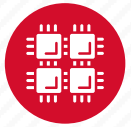
- Command to submit a job
  - **`qsub` *`script_file`***
- Response from PBS (example)
  - `123456.owens-batch.ten.osc.edu`
- Show status of batch jobs
  - **`qstat -a` *`jobid`***
  - **`qstat -u` *`username`***
  - **`qstat -f` *`jobid`***
- Delete a batch job
  - **`qdel` *`jobid`***

List of Batch commands on osc.edu



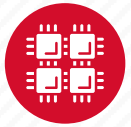
Scheduling Policies and Limits

- 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

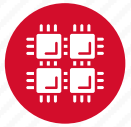
- Queue wait time depends on many factors
  - System load
  - Resources requested
    - nodes, cores, large memory, gpus, software licenses
  - Fair share limits (if load is high)
    - reduced priority for users or groups using a lot of resources



Job Output

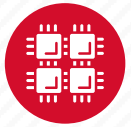
- Screen output ends up in file ***job_name.ojobid***
  - Copied to your working directory when job ends
  - Example: **testjob.o1234567**
- To see screen output while job is running
  - **qpeek *jobid***
  - Example: **qpeek 1234567**

```
----------------------
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
----------------------
```



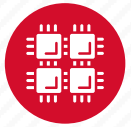
Interactive Batch Jobs

- 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`



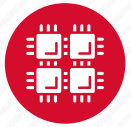
Batch Queues

- 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`



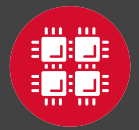
Parallel Computing

- 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