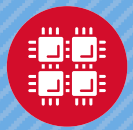
Ohio Supercomputer Center

An OH·TECH Consortium Member

Computing Services to Accelerate Research and Innovation

An introduction to OSC services, hardware, and environment



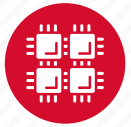
Kate Cahill

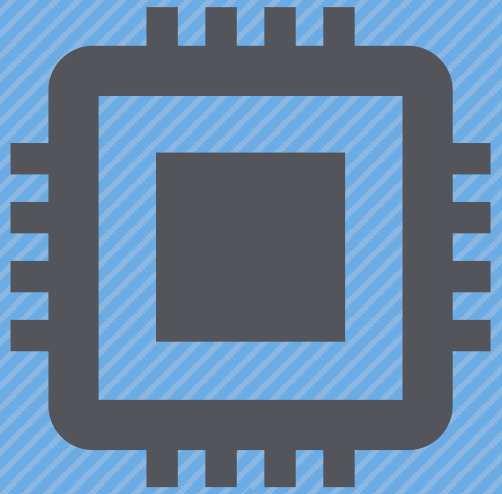
Education & Training Specialist

“OSC is here to empower your research.”

Outline

- What is OSC?
- HPC Concepts
- Hardware Overview
- Data Storage Systems
- Batch Processing
- Accessing Available Software
- OSC OnDemand Web Portal Demonstration



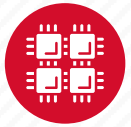


What is the Ohio Supercomputer Center?

"640K ought to be enough for anybody." – Not Bill Gates

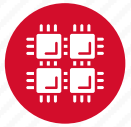
About OSC

- 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



HPC 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



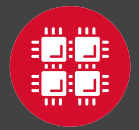
What is the difference between your laptop and a supercomputer?

X 10000 =

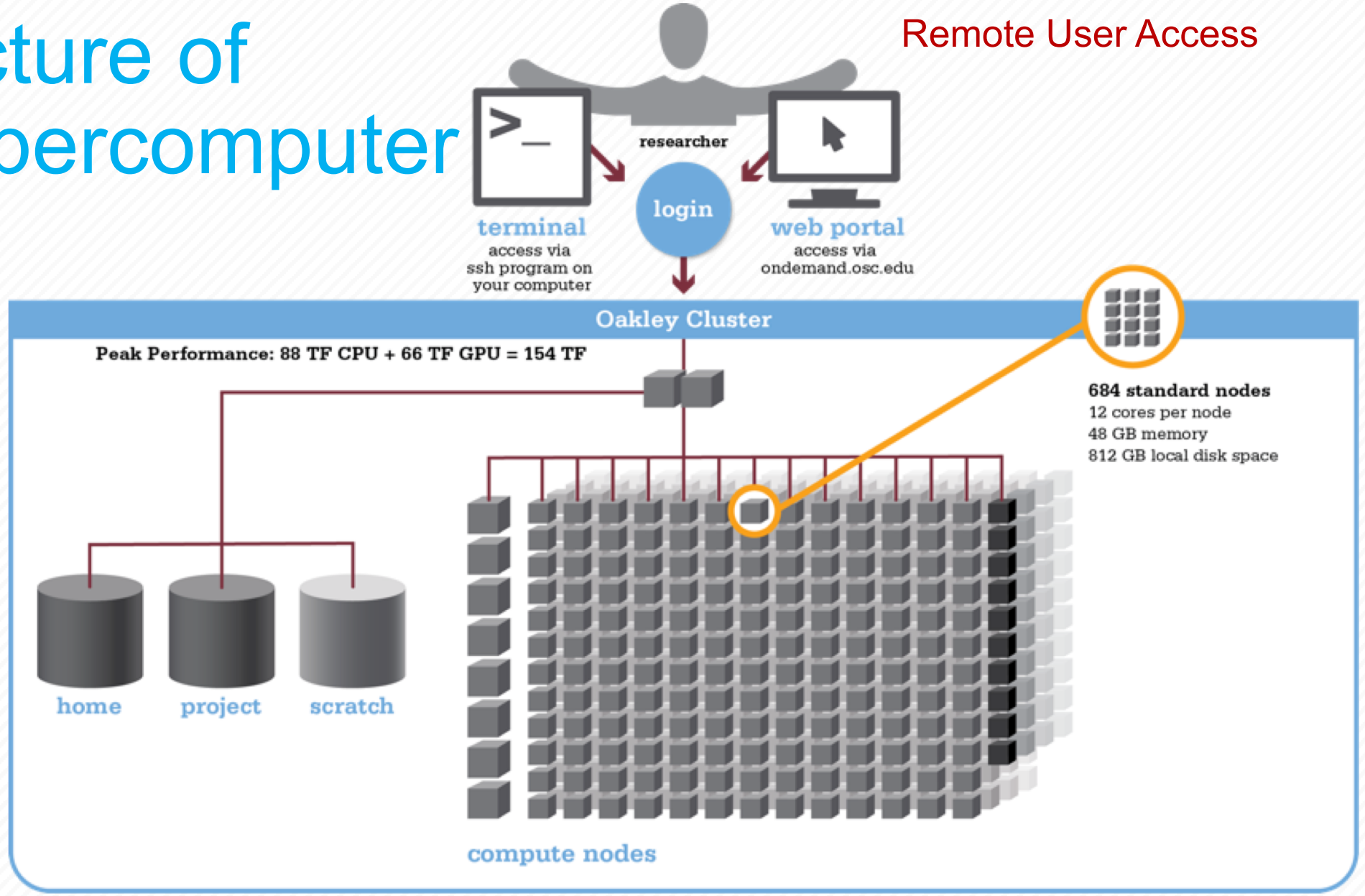
Remote Connect

HPC Terminology

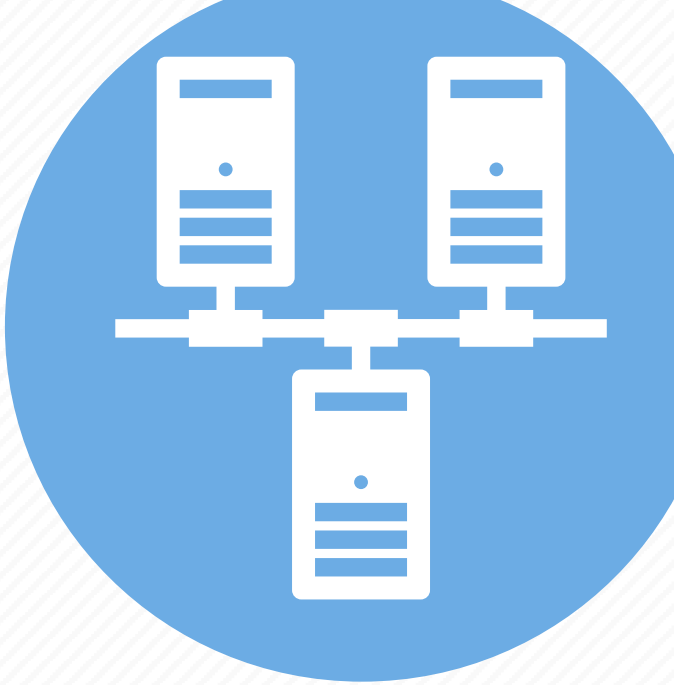
- Cluster
  - A group of computers (nodes) connected by a high-speed network, forming a supercomputer
- Node
  - Equivalent to a high-end workstation, part of a cluster
- Core
  - A processor (CPU), multiple cores per processor chip
- FLOPS
  - “FLoating-point Operations (calculations) Per Second”
-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



Structure of a Supercomputer



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!!

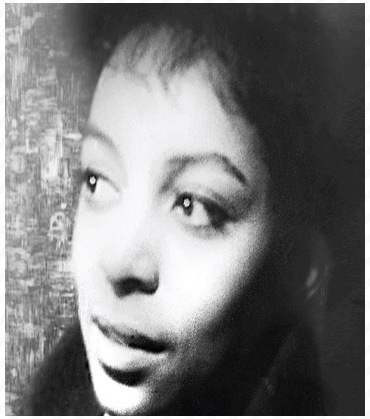
- Longer-term or archive

Hardware Overview

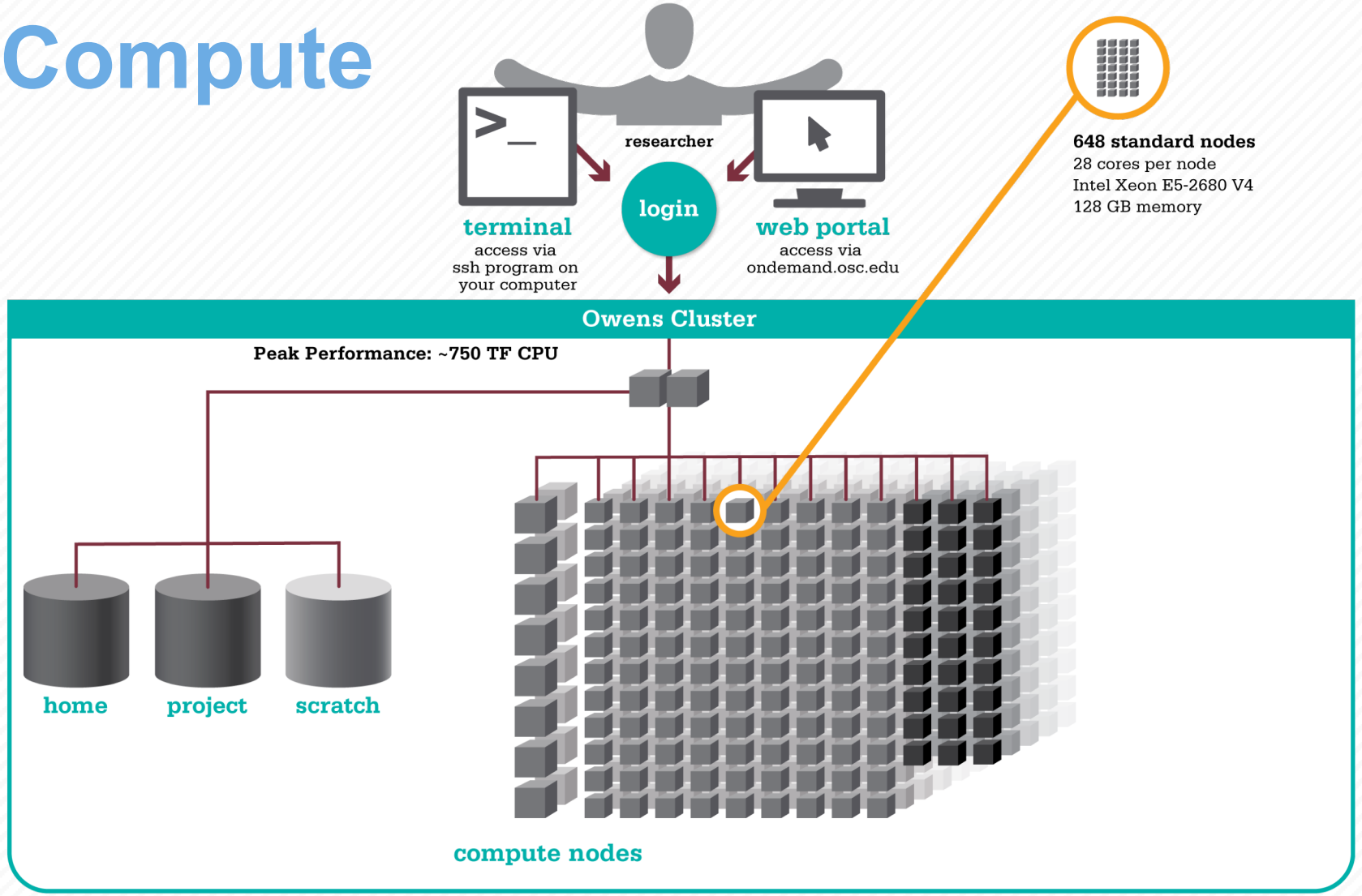
“To err is human, but to really foul things up you need a computer.” – Paul Ehrlich

System Configurations

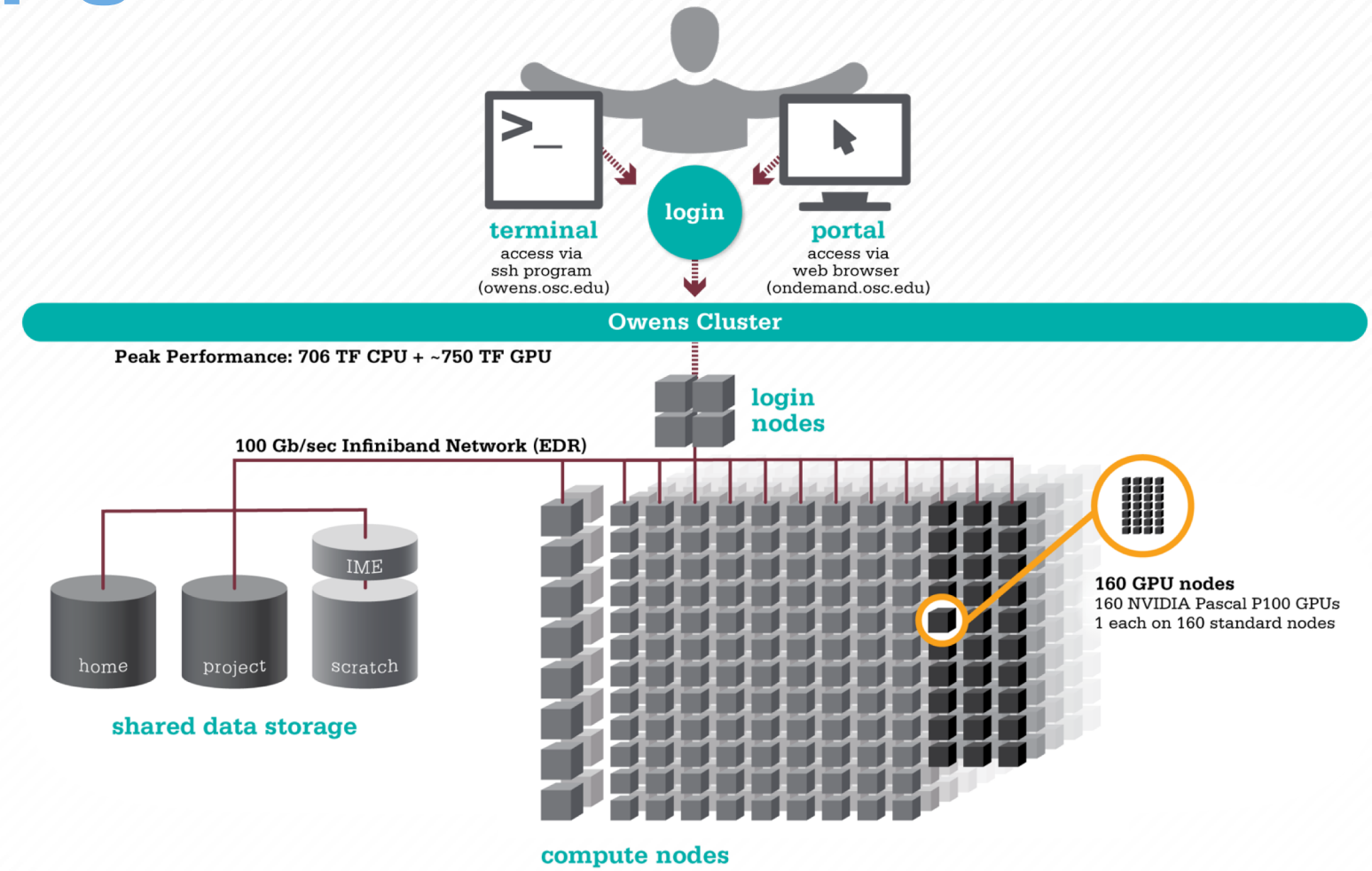
	Pitzer (2018)	Owens (2016)	Ruby (2014)
Theoretical Performance	~1300 TF	~1600 TF	~144 TF
# Nodes	260	824	240
# CPU Cores	10,560	23,392	4,800
Total Memory	~70.6 TB	~120 TB	~15.3 TB
Memory per Core	>5 GB	>5 GB	3.2 GB
Interconnect	EDR IB	EDR IB	FDR/EN IB



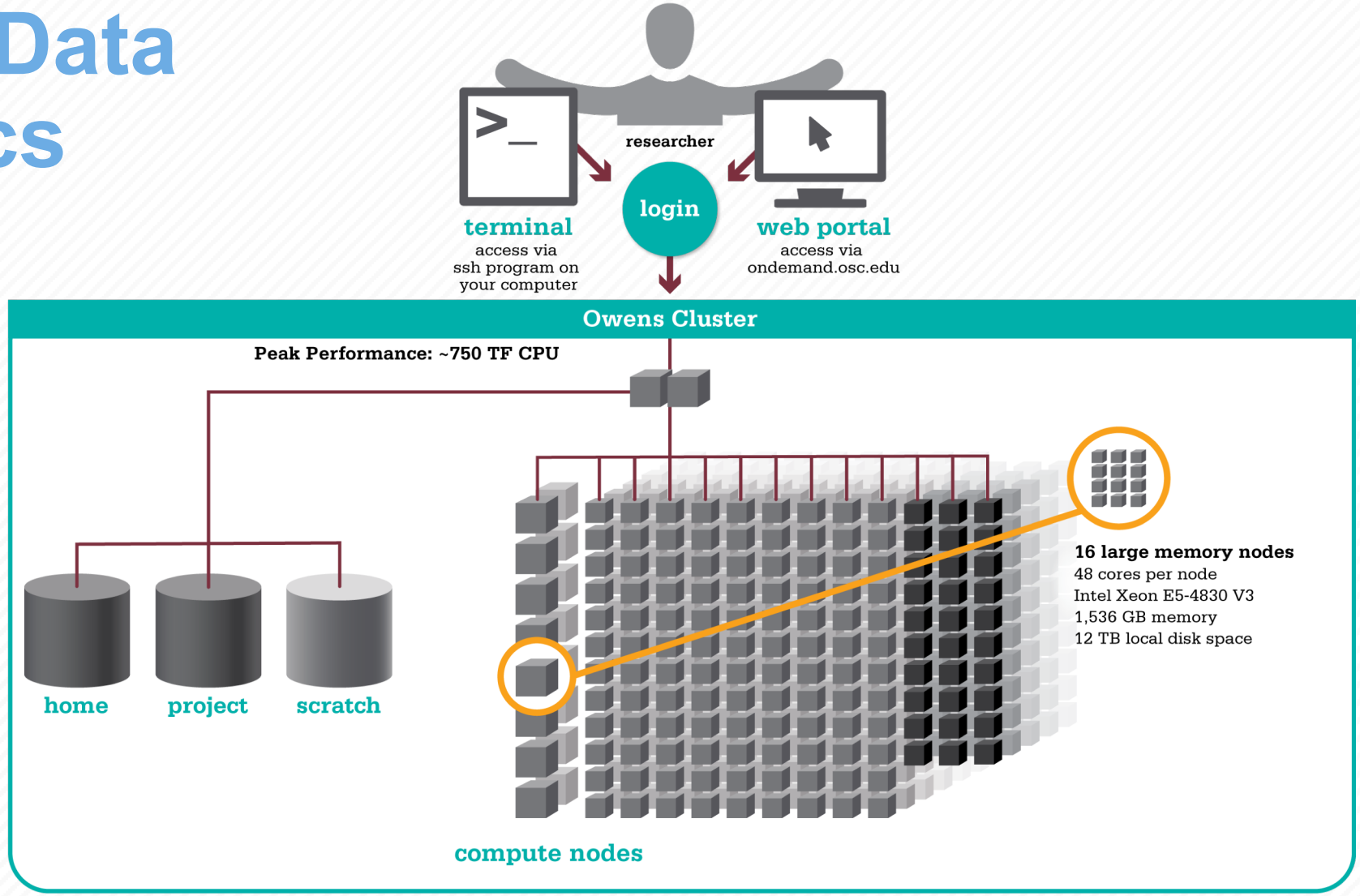
Owens Compute Nodes



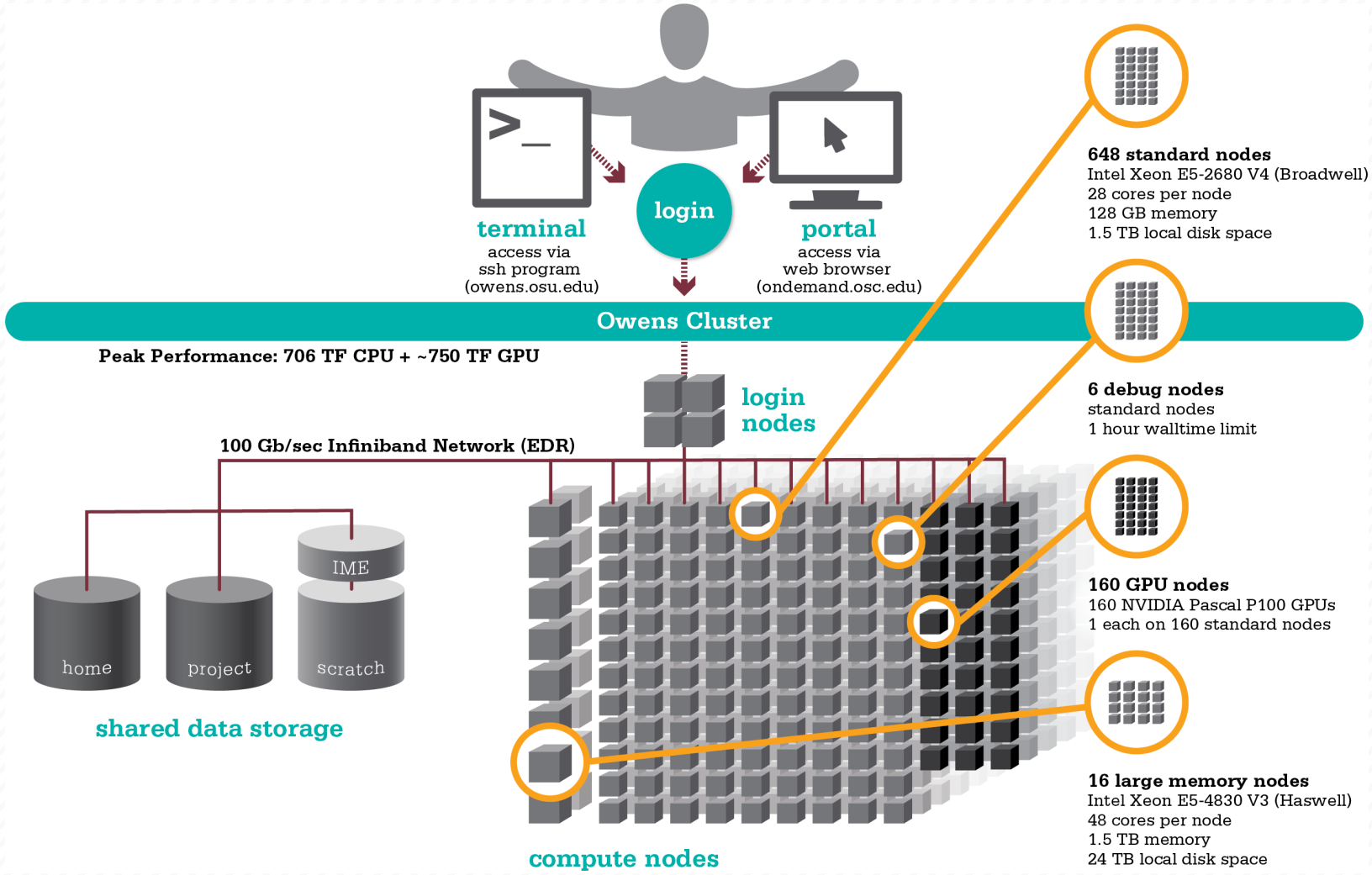
Owens GPU Nodes



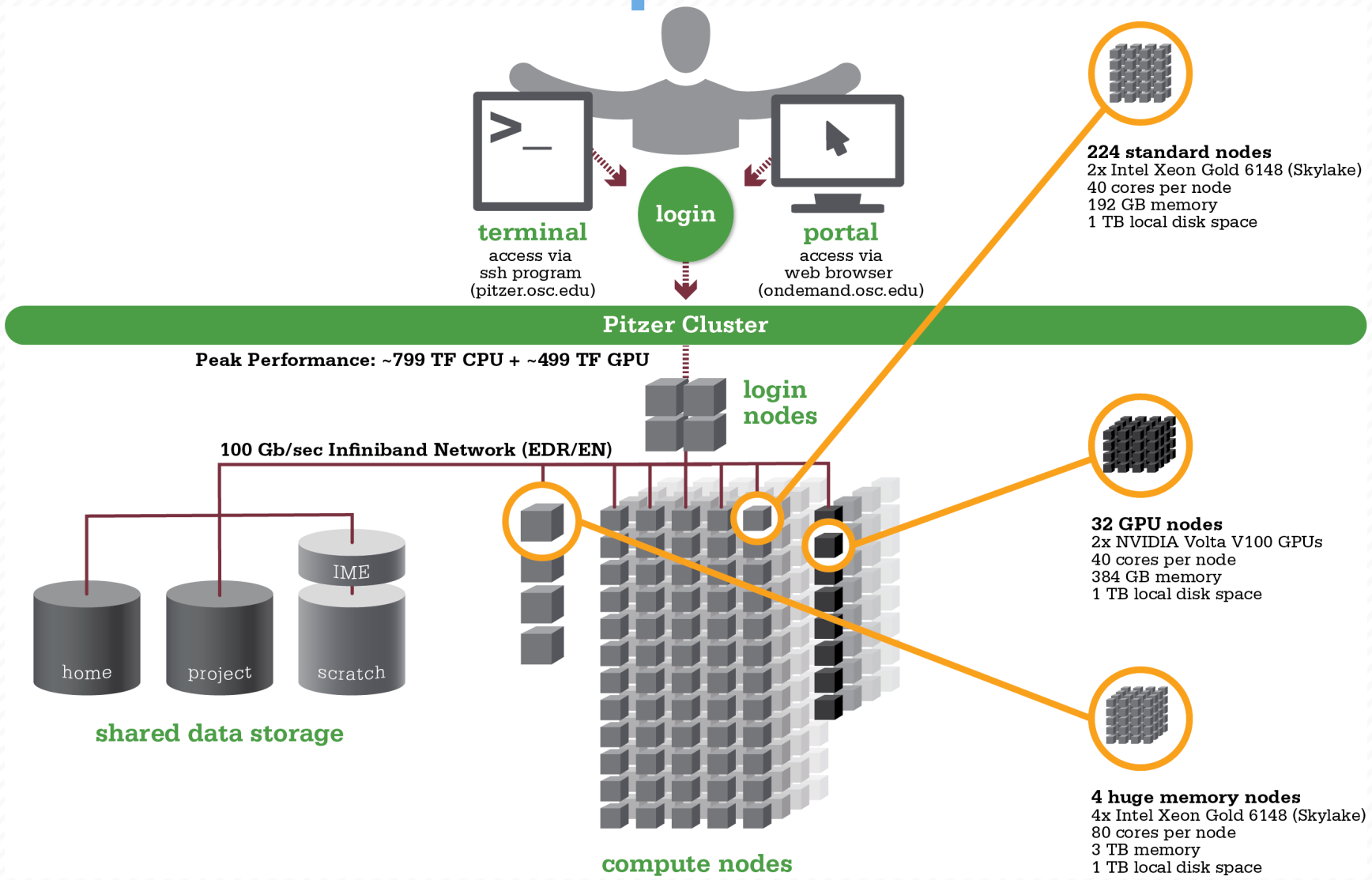
Owens Data Analytics Nodes



Owens Cluster Specifications

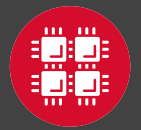


Pitzer Cluster Specifications



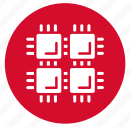
Login Nodes – Usage

- 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!**

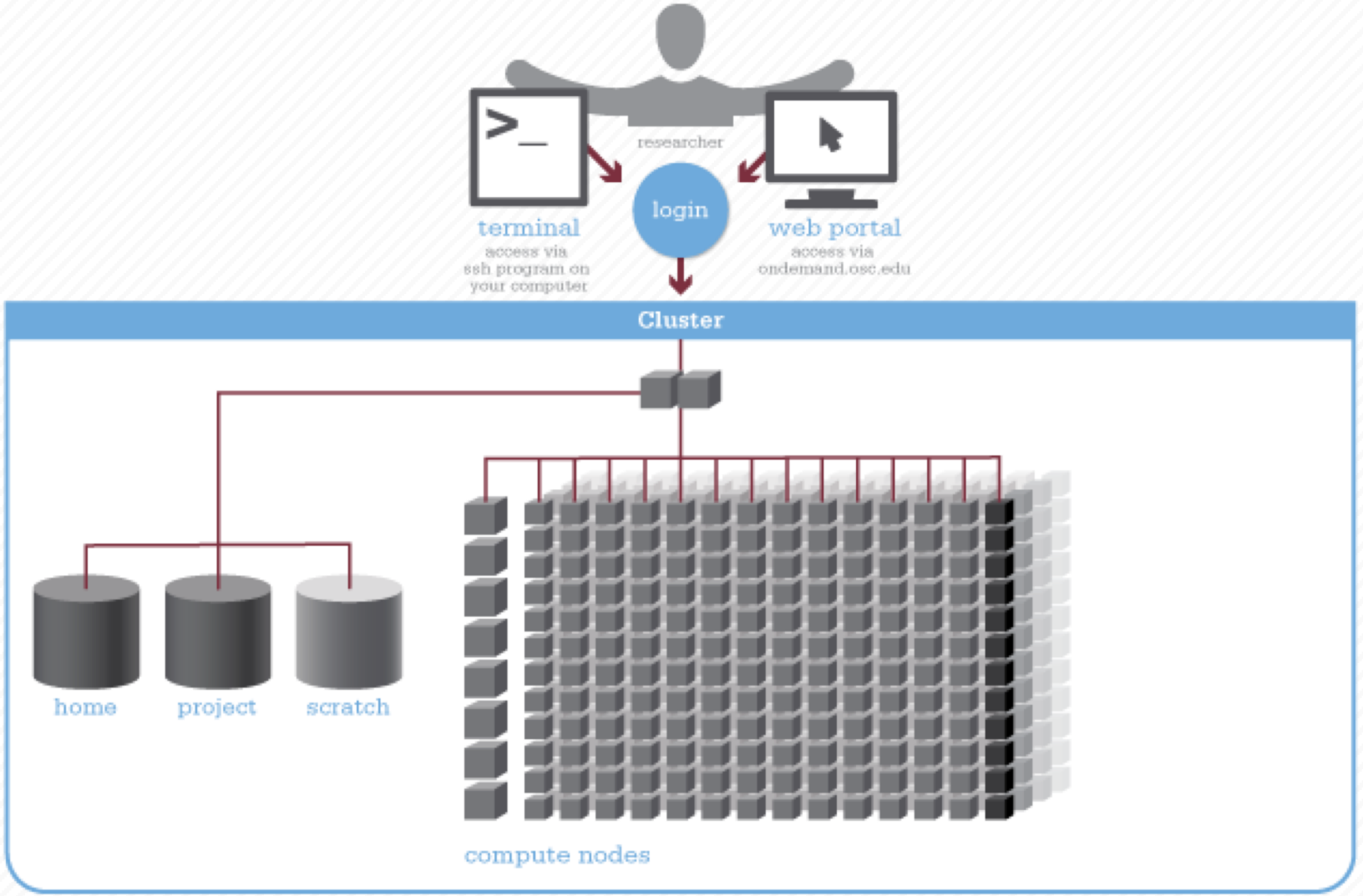


Data Storage Systems

”War is ninety percent information.” – Napoleon Bonaparte

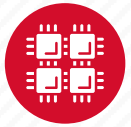


Four different file systems

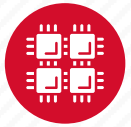


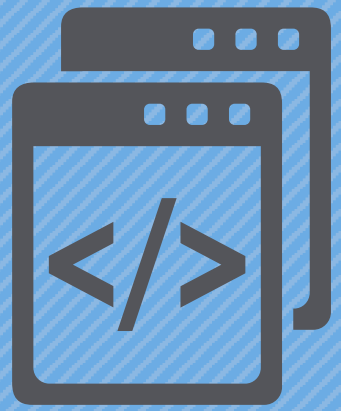
Filesystem Overview

- Home
  - Store your files here, backed up daily
  - Use $HOME or ~*username* to reference location
- Project
  - Available to Project PIs by request; shared by all users on a project, backed up daily
  - Use /fs/project/*project#* to reference location
- Scratch
  - Store large input or output files here
  - Faster I/O than Home or Project
  - Temporary storage, not backed up
- $TMPDIR
  - Storage on compute nodes, for use during your batch job
  - Be sure to copy any results back to Home at the end of your job, all data purged when job quits



Filesystem	Quota	Backed-Up?	Purged?
Home ($HOME)	500GB	Yes	No
Project (/fs/project)	By request	Yes	No
Scratch (/fs/scratch)	None	No	Yes – 120 days
Compute ($TMPDIR)	800GB (Oakley), 1 TB (Ruby & Owens)	No	Yes – when job completes



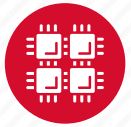


Getting Started at OSC

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

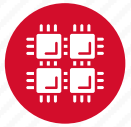
My.osc.edu

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



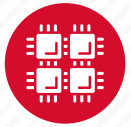
Your Contact Info

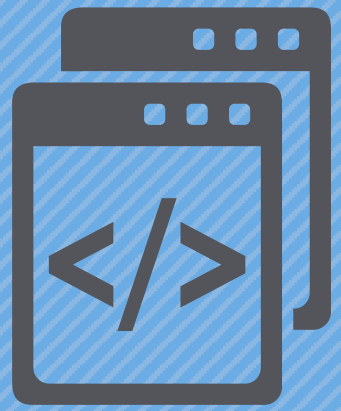
- Keep your contact information current
  - Use my.osc.edu to manage your account details.
- If your student continues to use OSC after graduation, make sure email address is updated
  - Acceptable if still collaborating with you
- May need to contact you about problems
- Will need to contact you about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



System Status

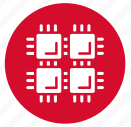
- Check system status on:
  - Message of the day (/etc/motd) – displayed at login
  - Twitter: @HPCnotices
  - Email for major outages or problems
- Scheduled downtimes
  - Quarterly maintenance – usually one day outage
  - Jobs held for up to two weeks prior
  - Next downtime is October 8





User Environment

“After growing wildly for years, the field of computing appears to be reaching its infancy.” – John Pierce



Linux Operating System

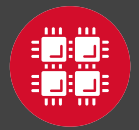
- “UNIX-like”
- Widely used in HPC
- Mostly command-line
- Choice of shells (bash is default)
- Freely distributable, open-source software
- Tutorials available
- www.linux.org

Connecting to the clusters

- Connect to OSC machines using **`ssh`** (secure shell)
  - From a Linux/UNIX machine : At prompt, enter

    `ssh` ***`userid`*****`@owens.osc.edu`**

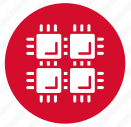
    `ssh` ***`userid`*****`@ruby.osc.edu`**
  - From a Mac: Enter **`ssh`** command in TERMINAL window
  - From Windows: **`ssh`** client software needed
    - Both commercial and free versions are available
- New: Connect using OSC OnDemand portal (web-based)



Connecting to an OSC Cluster with Graphics

- Programs on the cluster can have an X-based GUI
  - Display graphics on your computer
- Linux/UNIX and Mac: Use **`-X`** flag

  `ssh -X` ***`userid`*****`@oakley.osc.edu`**
- Windows: Need extra software
  - Both commercial and free versions are available
  - Configure your ssh client to tunnel or forward X11
- Primarily used with programs on login node
  - Can also use with interactive batch jobs



OSC OnDemand
ondemand.osc.edu

- 1: User Interface
  - Web based
    - Usable from computers, tablets, smartphones
    - Zero installation
  - Single point of entry
    - User needs three things
      - ondemand.osc.edu
      - OSC Username
      - OSC Password

    Connected to all resources at OSC

- 2: Interactive Services
  - File Access
  - Job Management
  - Visualization Apps
    - Desktop access
    - Single-click apps (Abaqus, Ansys, Comsol, Paraview)
  - Terminal Access

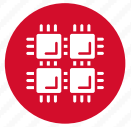
Tutorial available at

osc.edu/ondemand

Transferring Files to and from the Cluster

- Most file transfers to and from OSC machines use **`sftp`** or **`scp`**
  - Linux and Mac have them built in
  - Windows needs extra software - FileZilla
- For small files, connect to a login node

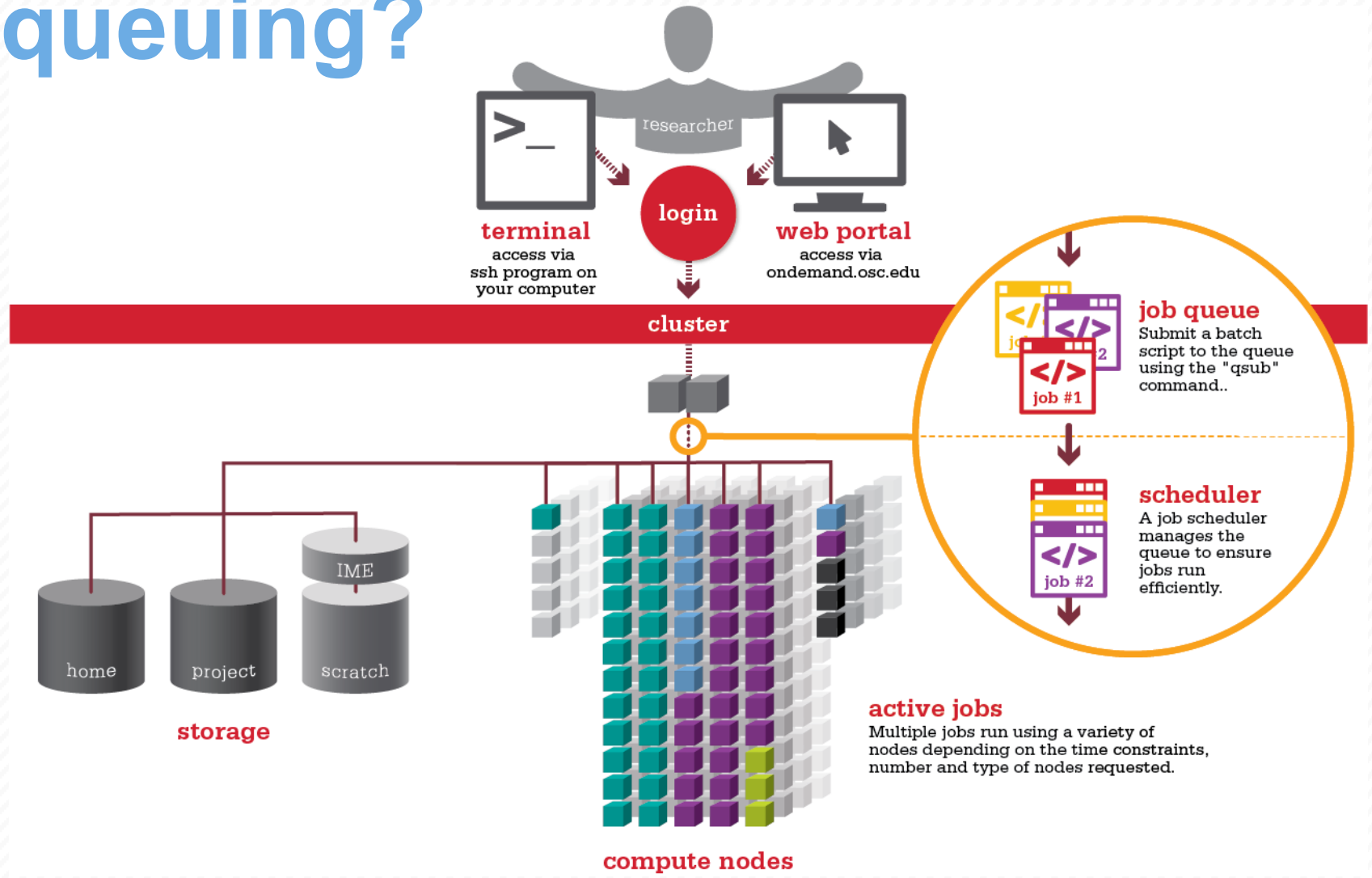
  `owens.osc.edu`
- For large files, transfer may fail due to shell limits
  - Connect to **`sftp.osc.edu`** (file transfer only)
- OnDemand drag and drop file transfer up to 5GB files



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

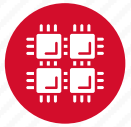
Why do supercomputers use queuing?



Batch System at OSC

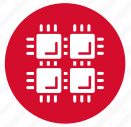
- Compute nodes are allocated through the batch system
  - PBS – Portable Batch System
  - Torque – resource manager
  - Moab – scheduler
- Documentation at

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



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

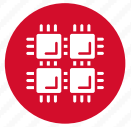


Steps for Running a Job on the Compute Nodes

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

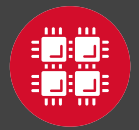
```
#PBS -N serial_fluent
#PBS -l walltime=1:00:00
#PBS -l nodes=1:ppn=28:gpus=1
#PBS -A PAS####
#PBS -j oe
#PBS -l software=fluent+1
# Set up the FLUENT environment
module load fluent
# Move to directory job was submitted from
cd $PBS_O_WORKDIR
# Copy input files to compute node
cp run.input $TMPDIR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

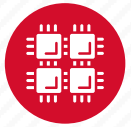
Put all this into a text file!



Submitting a Job and Checking Status

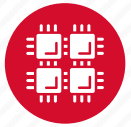
- Command to submit a job
  - **`qsub` *`script_file`***
- Response from PBS (example)
  - `123456.owens-batch.ten.osc.edu`
- Show status of batch jobs
  - **`qstat -a` *`jobid`***
  - **`qstat -u` *`username`***
  - **`qstat -f` *`jobid`***
- Delete a job
  - ***`qdel jobid`***

List of Batch commands on osc.edu



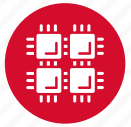
Scheduling Policies and Limits

- 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

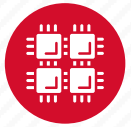
- Queue wait time depends on many factors
  - System load
  - Resources requested
    - nodes, cores, large memory, gpus, software licenses
  - Fair share limits (if load is high)
    - reduced priority for users or groups using a lot of resources



Job Output

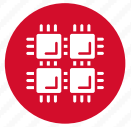
- Screen output ends up in file ***job_name.ojobid***
  - Copied to your working directory when job ends
  - Example: **testjob.o1234567**
- To see screen output while job is running
  - Job log is updated throughout job in working directory, **cat** or **tail *job_name.ojobid***

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



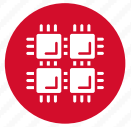
Interactive Batch Jobs

- 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`



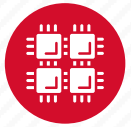
Batch Queues

- 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`**



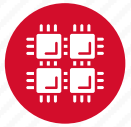
Parallel Computing

- 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

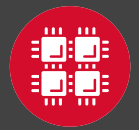
- Program must be written to take advantage of multiple cores and/or multiple nodes
- Many commercial applications have multithreaded or parallel versions
- Must use `mpiexec` for multiple nodes
- **Can't just request more nodes or cores and expect your job to run faster**



Loading and Running Software

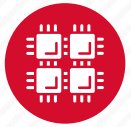
Software Maintained by OSC

- 145 software packages maintained for users
- Always first check software page on https://www.osc.edu/resources/available_software
  - Version information for all clusters
  - License information – some software you must request access
  - Usage examples



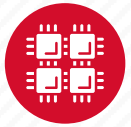
Modules for Software access

- How modules work
  - Modify environment variables like **`$PATH`** and **`$MANPATH`** within your shell
- Default set of modules loaded at login
  - module system, batch system (do not unload)
  - default compiler and MPI modules
- Do NOT completely replace **`$PATH`** in your **`.bash_profile`** or **`.bashrc`**
- DO prepend directories to the existing **`$PATH`**
  - Type: **`export PATH=$HOME/bin:$PATH`**



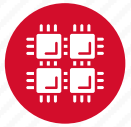
Adding or Removing Software from Your Environment

- Load the module for the software you need, e.g.,
  - `module load comsol`
- Allows multiple versions of software to coexist on our system
- Allow us to make changes without affecting you
  - PLEASE DON'T HARDCODE PATHS!
- Can load modules at command prompt or in your `.bash_profile` or `.bashrc` file
- Also load modules in your job (batch) scripts



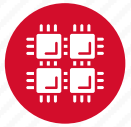
Module Commands

- What modules do you have loaded?
  - `module list`
- What modules are available?
  - `module spider` or `module avail`
- Multiple versions of the same software
  - `module avail intel`
- Add a software module to your environment
  - `module load cuda`
- Remove a software package from your environment
  - `module unload intel`
- Load a different software version
  - `module swap intel intel/13.1.3.192`



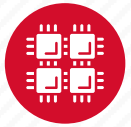
Third party applications

- **General programming software (**$statewide licensed)
  - gnu compilers and debugger
  - $ Intel compilers
  - $ Totalview debugger
  - $ Allinea profiler
  - MPI library
  - HDF5
  - NetCDF
  - Java, Java Virtual Machine
  - Python



Third party applications

- **Parallel programming software (₵**statewide licensed)
  - MPI library (mvapich, mvapich2)
  - OpenMP
  - CUDA
  - OpenCL
  - OpenACC

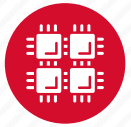


Access to Licensed Software

- Most software licenses for academic use only
- Some software requires signed license agreement
  - Check website
  - Contact OSC Help

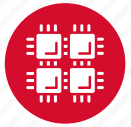
- List of applications can be found at Software page:

  http://www.osc.edu/supercomputing/software/



OSC doesn't have the software you need?

- Commercial software
  - Fill out a request form (see our FAQ)
  - SUG will consider it
- Open-source software
  - You can install it yourself in your home directory, see HOWTO
  - If there's enough demand, we can install it for shared use
- Have your own license?
  - Contact OSC Help



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

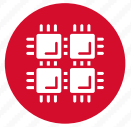
Installing Software

Installing R packages

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

System updates

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




Questions?

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