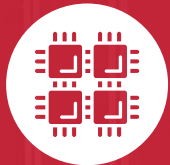


OWENS



JESSE OWENS
OLYMPIC CHAMPION, BEACON FOR EQUALITY, YOUTH ADVOCATE

 Ohio Supercomputer Center
An OH·TECH Consortium Member



Ohio Supercomputer Center

An OH·TECH Consortium Member

Computing Services to Accelerate Research and Innovation

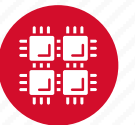
An introduction to OSC services, hardware, and environment





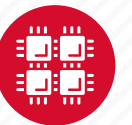
Kate Cahill
Education & Training Specialist

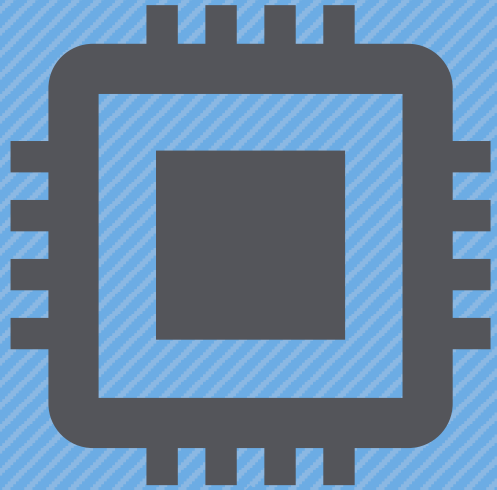
“OSC is here to empower your research.”



Outline

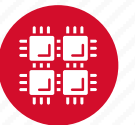
- What is OSC?
- HPC Concepts
- Hardware Overview
- Data Storage Systems
- Getting a New Project/Account
- Batch Processing
- Accessing Available Software
- OSC OnDemand Web Portal Demonstration





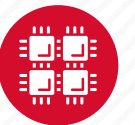
What is the Ohio Supercomputer Center?

"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



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.



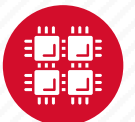
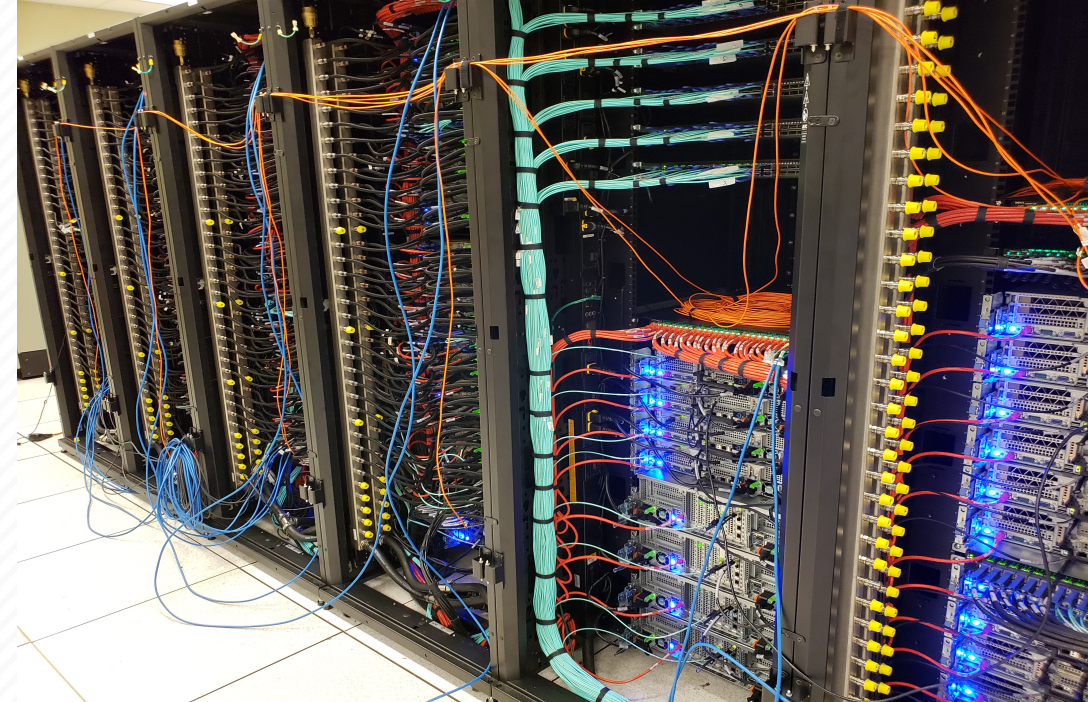
Web Software Development

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.



Utilization 2018

CY2018



226,600,000+
core-hours
consumed



85% average
HPC system
utilization



3,200,000+
computational
jobs



3,044 TF of
computational
power available



99.2%
up-time



59% average
storage system
utilization



3+ PB
data stored



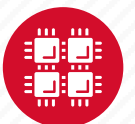
2 PB data
transferred



74% jobs
started within
30 minutes

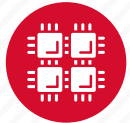
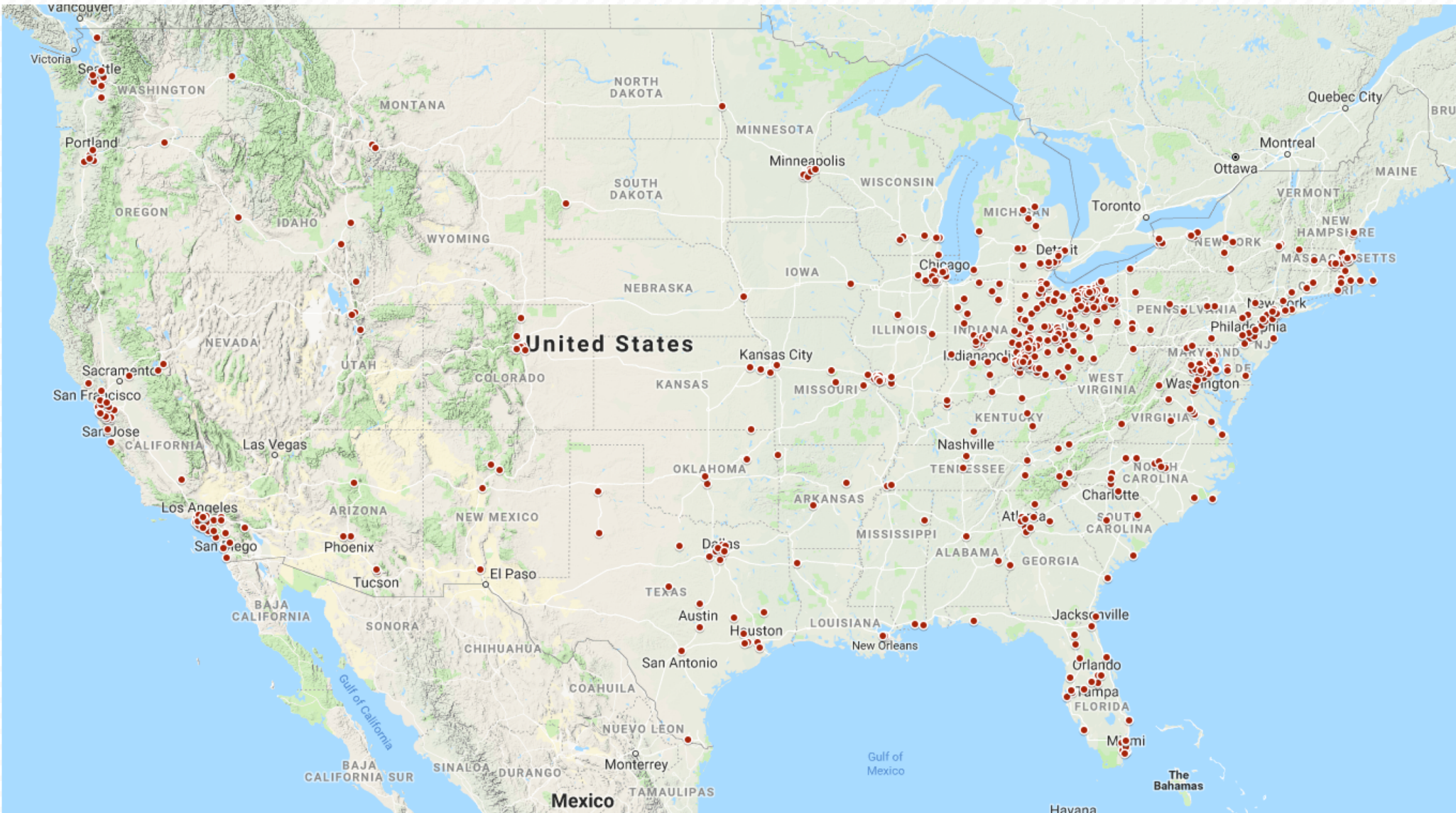


145 software
packages



OSC OnDemand Users

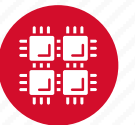
CY2018





HPC Usage and Case Studies

“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



Client Services 2018

CY2018



23 Ohio universities



56 companies



36 universities outside of Ohio



3,185 researchers



301 new projects created



315 trainees



693 projects served



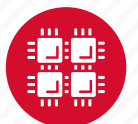
48 college courses used OSC



22 training opportunities

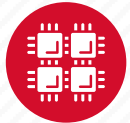
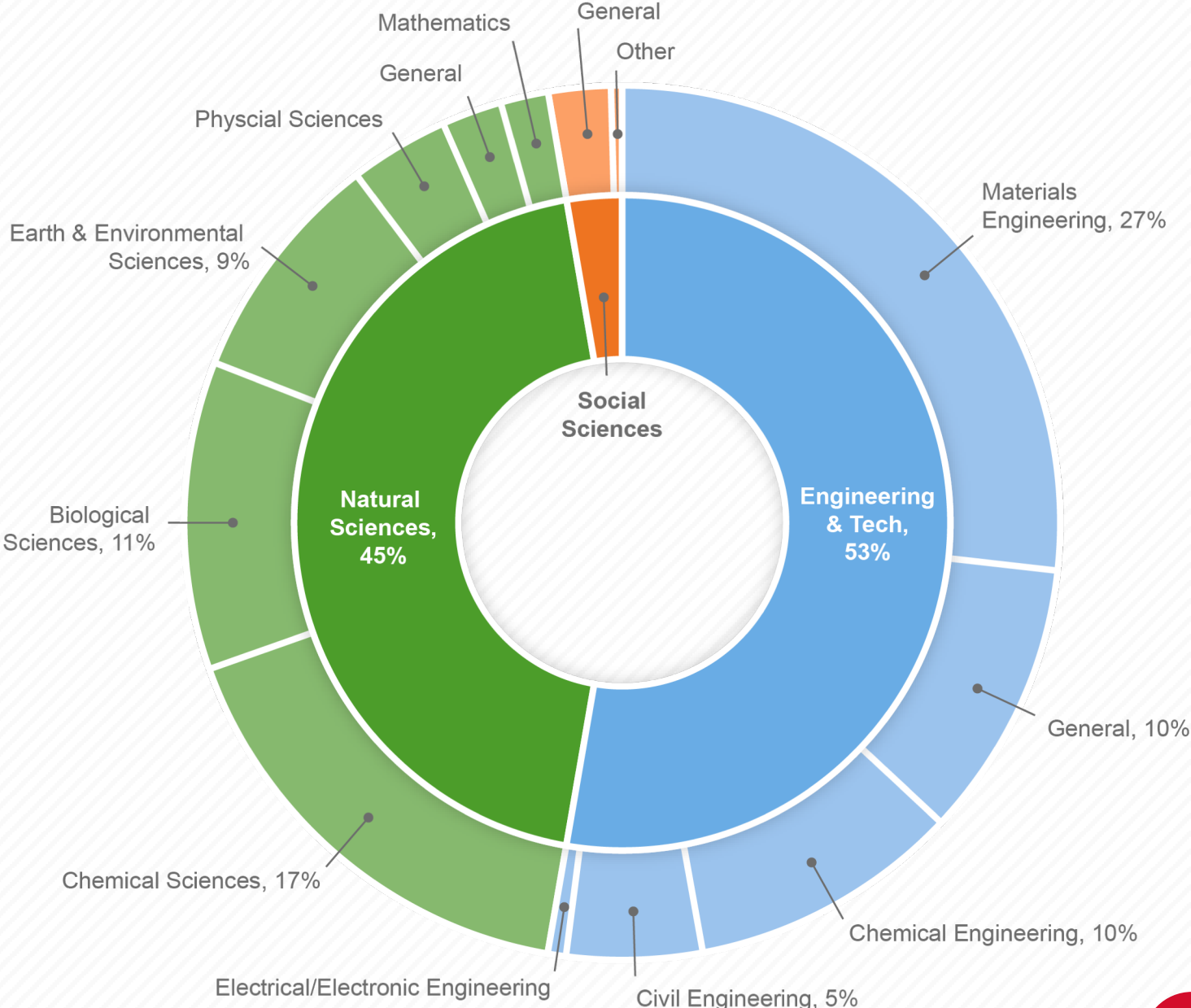


72 publications cited OSC



Usage by Field of Science

CY2017



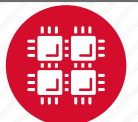
Classroom Usage

CY2018

Department	Users
BGSU Computer Science	23
Cedarville Engineering	14
Cedarville Engineering and Computer Science	8
Cleveland State Electrical Engineering and Computing Systems	8
Cleveland State Physics	8
Kenyon Chemistry	24
Miami Computer Science and Software Engineering	3
OSU Biomedical Engineering	13
OSU Biomedical Informatics	15
OSU Biomedical Informatics	0
OSU Chemical and Biomolecular Engineering	10
OSU Chemical and Biomolecular Engineering	24
OSU Chemistry	91
OSU Chemistry	129
OSU Chemistry	406
OSU Chemistry	170
OSU Chemistry and Biochemistry	10
OSU Chemistry and Biochemistry	14
OSU Computer Science and Engineering	24
OSU Computer Science and Engineering	3
OSU Computer Science and Engineering	2
OSU Computer Science and Engineering	33
OSU Computer Science and Engineering	73
OSU Computer Science and Engineering	17

Department	Users
OSU Computer Science and Engineering	60
OSU Evolution, Ecology and Organismal Biology	13
OSU Evolution, Ecology and Organismal Biology	23
OSU Evolution, Ecology and Organismal Biology	16
OSU Geography	9
OSU Horticulture and Crop Science	12
OSU Linguistics	11
OSU Materials Science and Engineering	4
OSU Materials Science and Engineering	20
OSU Mechanical and Aerospace Engineering	4
OSU Mechanical and Aerospace Engineering	10
OSU Physics	7
OSC Training	7
U. Cincinnati Biology	19
U. Cincinnati Chemistry	8
U. Cincinnati Electrical Engineering and Computing Systems	11
U. Cincinnati Electrical Engineering and Computing Systems	24
U. Cincinnati Electrical Engineering and Computing Systems	53
U. Cincinnati Electrical Engineering and Computing Systems	46
U. Cincinnati Physics	49
U. Dayton Mathematics	3
Wright State Psychology	11
Xavier Mathematics and Computer Science	20
Xavier Physics	11

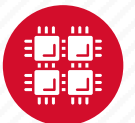
10 Universities. 26 Departments. 48 courses. 1,573 students.





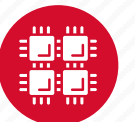
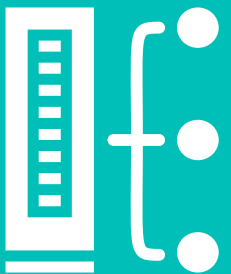
HPC Concepts

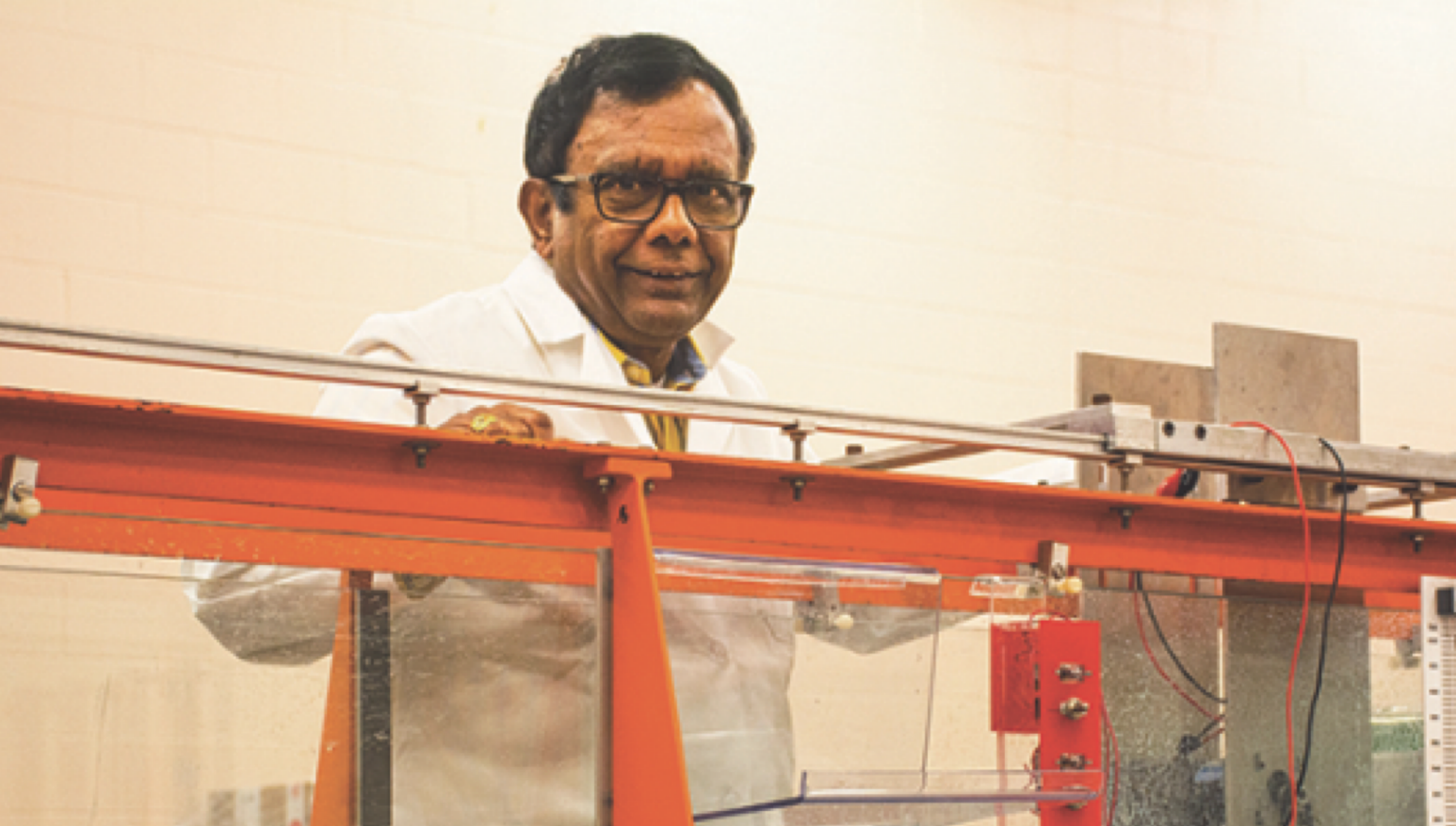
“The difference between us and a computer is that, the computer is blindingly stupid, but it is capable of being stupid many, many million times a second.” – Douglas Adams



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

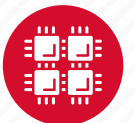




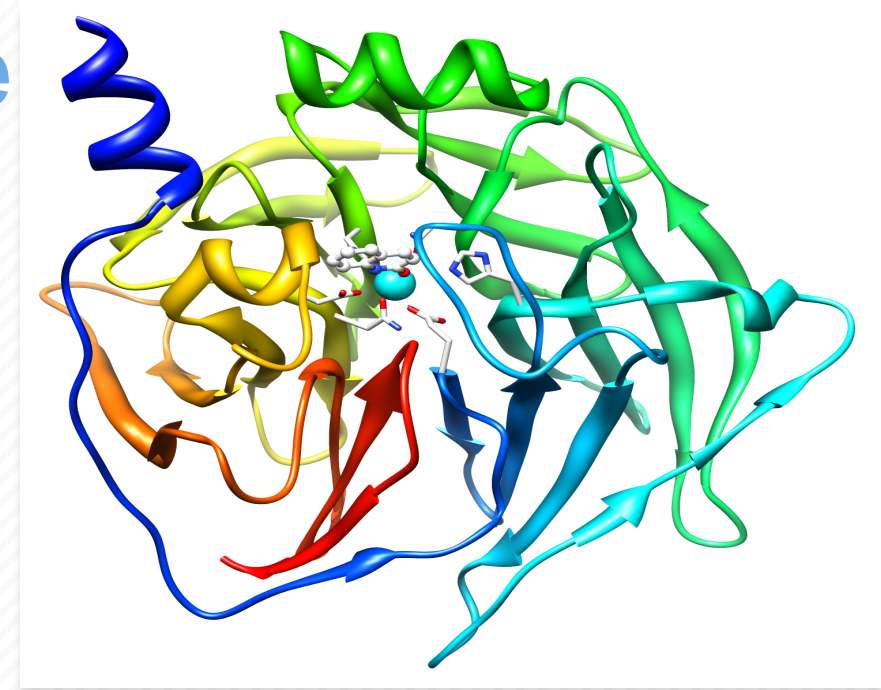
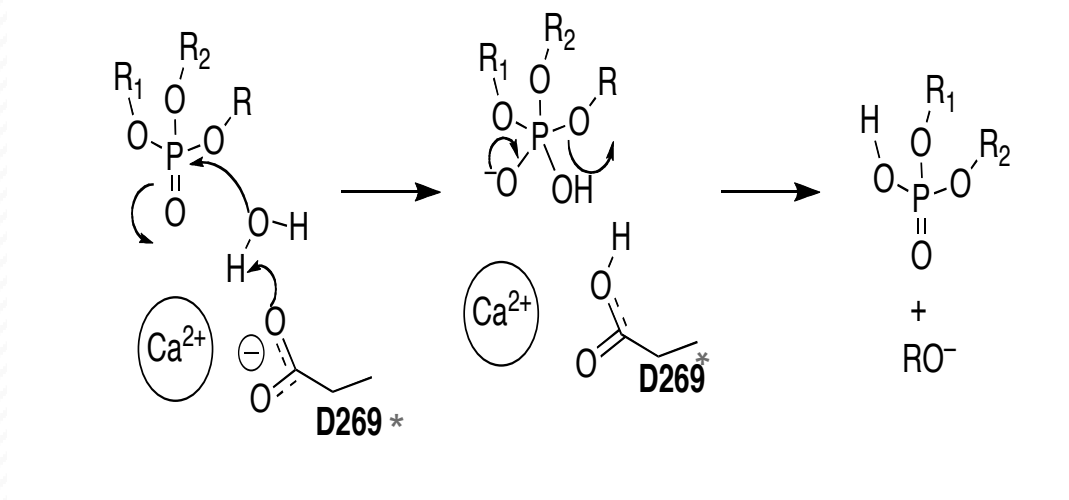
Renewable Energy

Researchers using computational fluid dynamics studies to design new hydropower turbines

PI: Subramania Sritharan, Ph.D., Associate Director for research, Land-Grant Program, Central State University Credit: WKMS



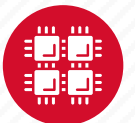
Treating Nerve Agent Exposure



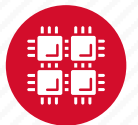
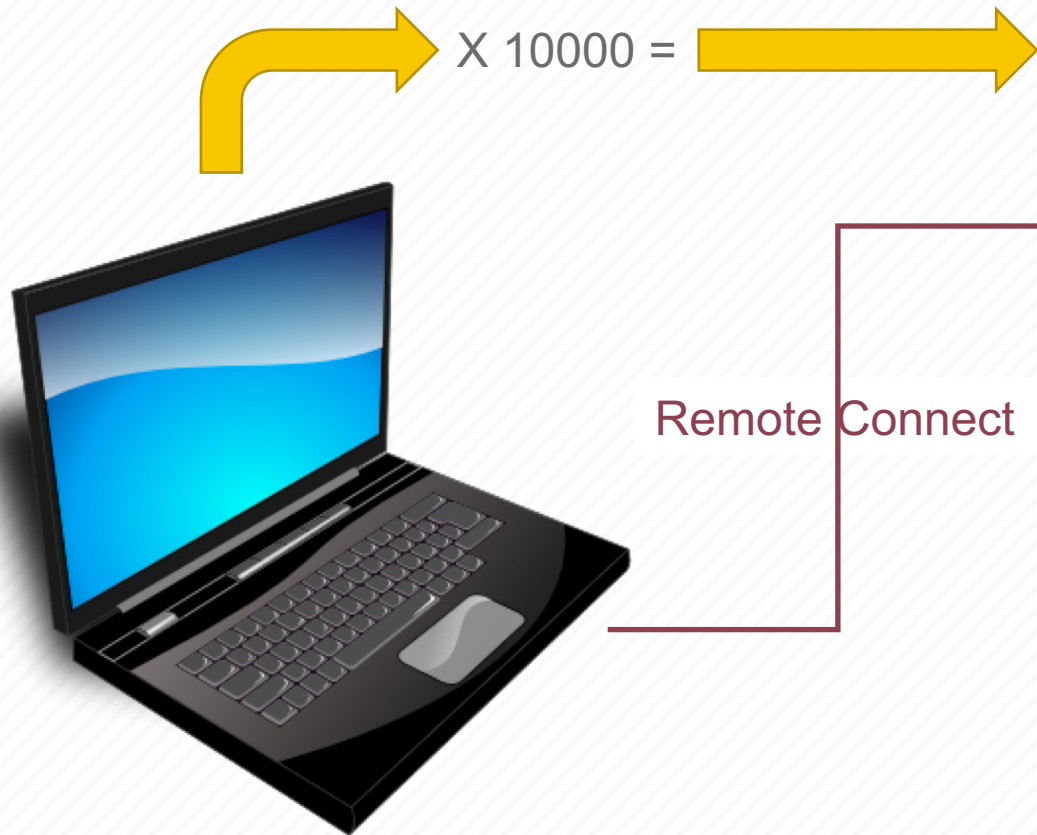
Treating Nerve Agent Exposure

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

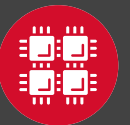


What is the difference between your laptop and a supercomputer?



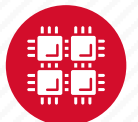
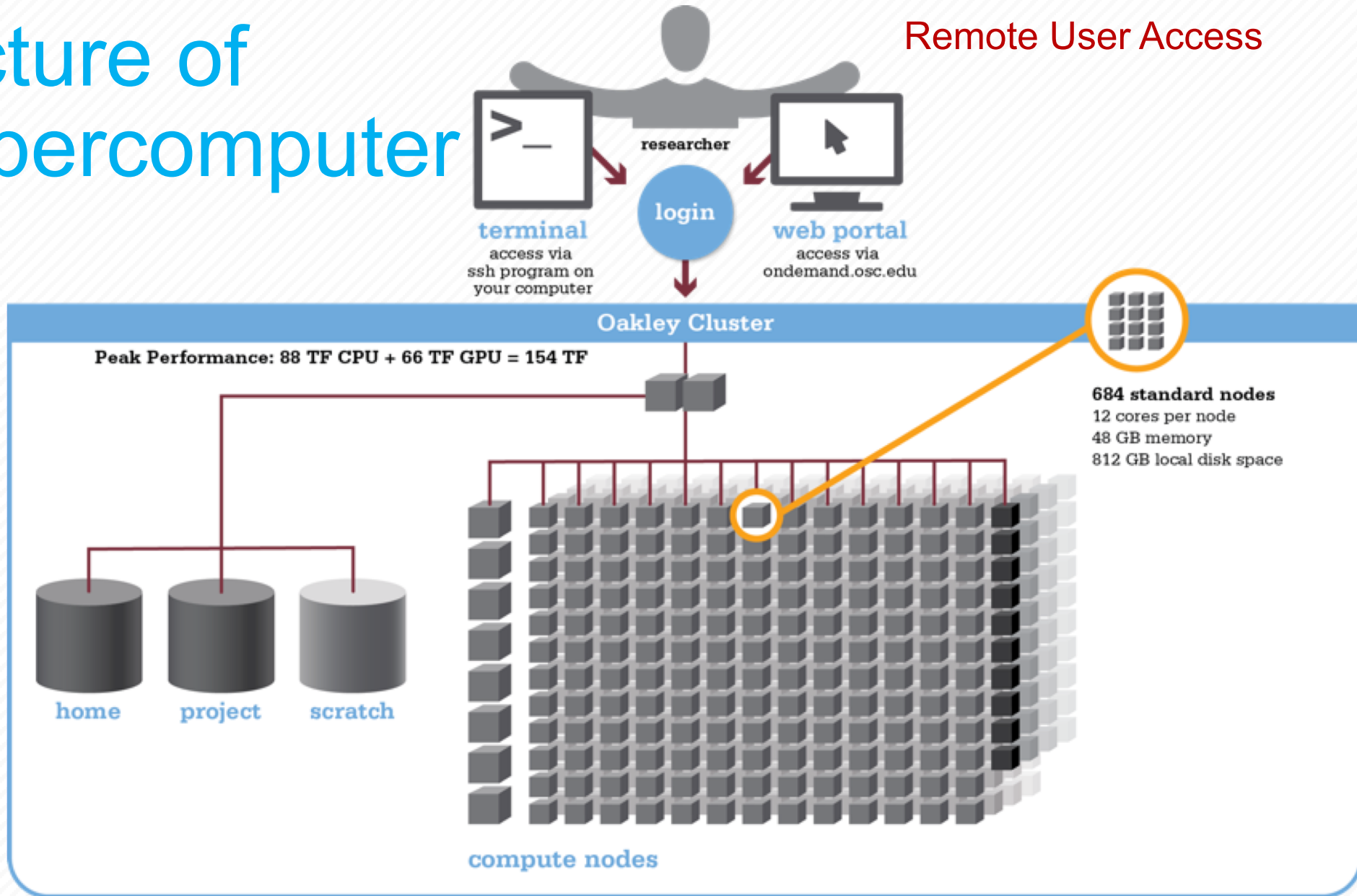
HPC Terminology

- 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

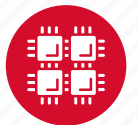


Memory

- 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

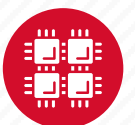
Parallel filesystems, eg Lustre or GPFS

Traditionally tuned for high bandwidth, not high IOPS

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

Short-term storage only!!

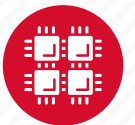
- Longer-term or archive





Hardware Overview

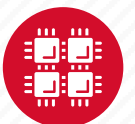
“To err is human, but to really foul things up you need a computer.” – Paul Ehrlich



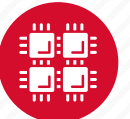
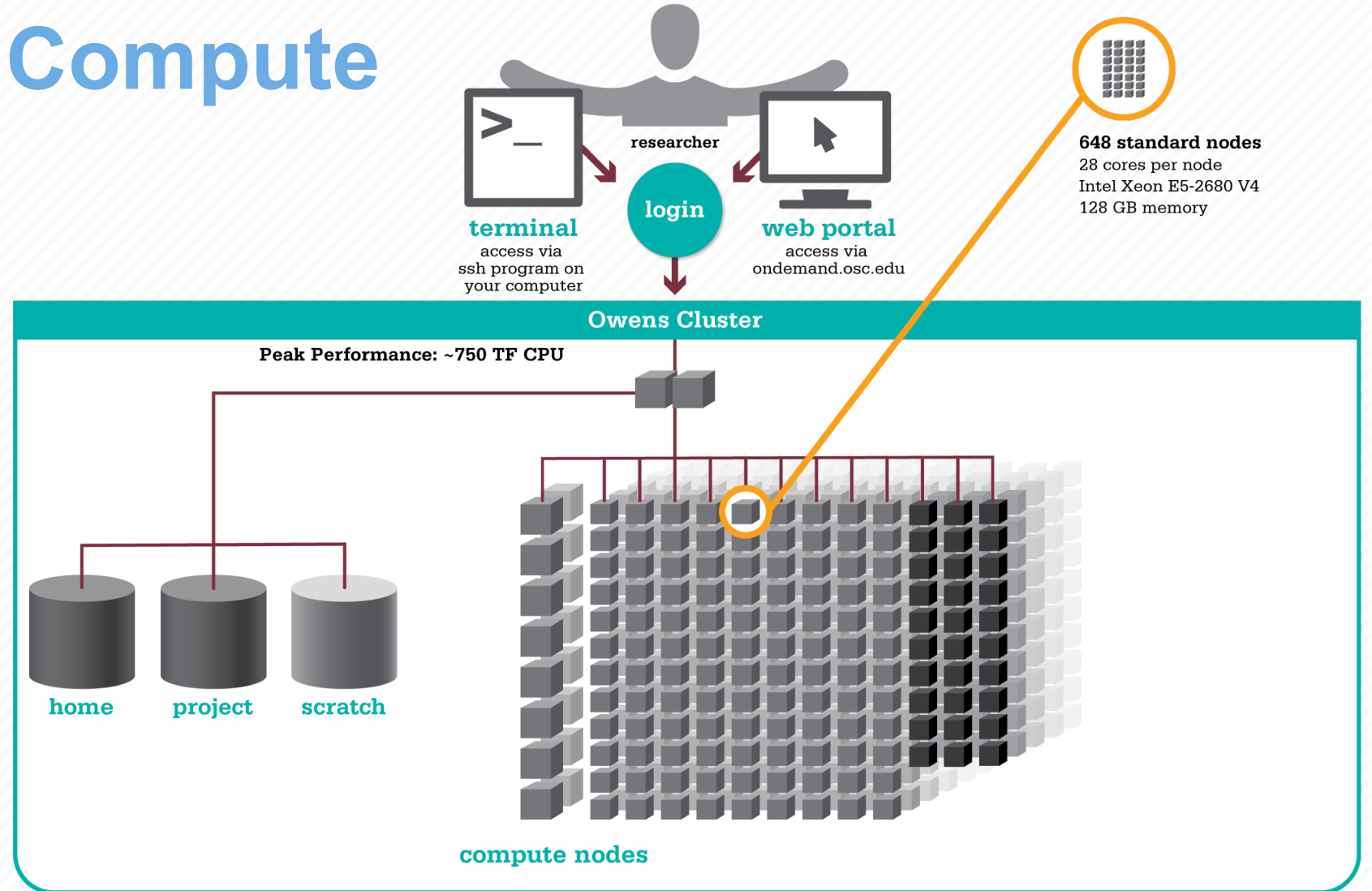
System Configurations



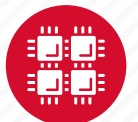
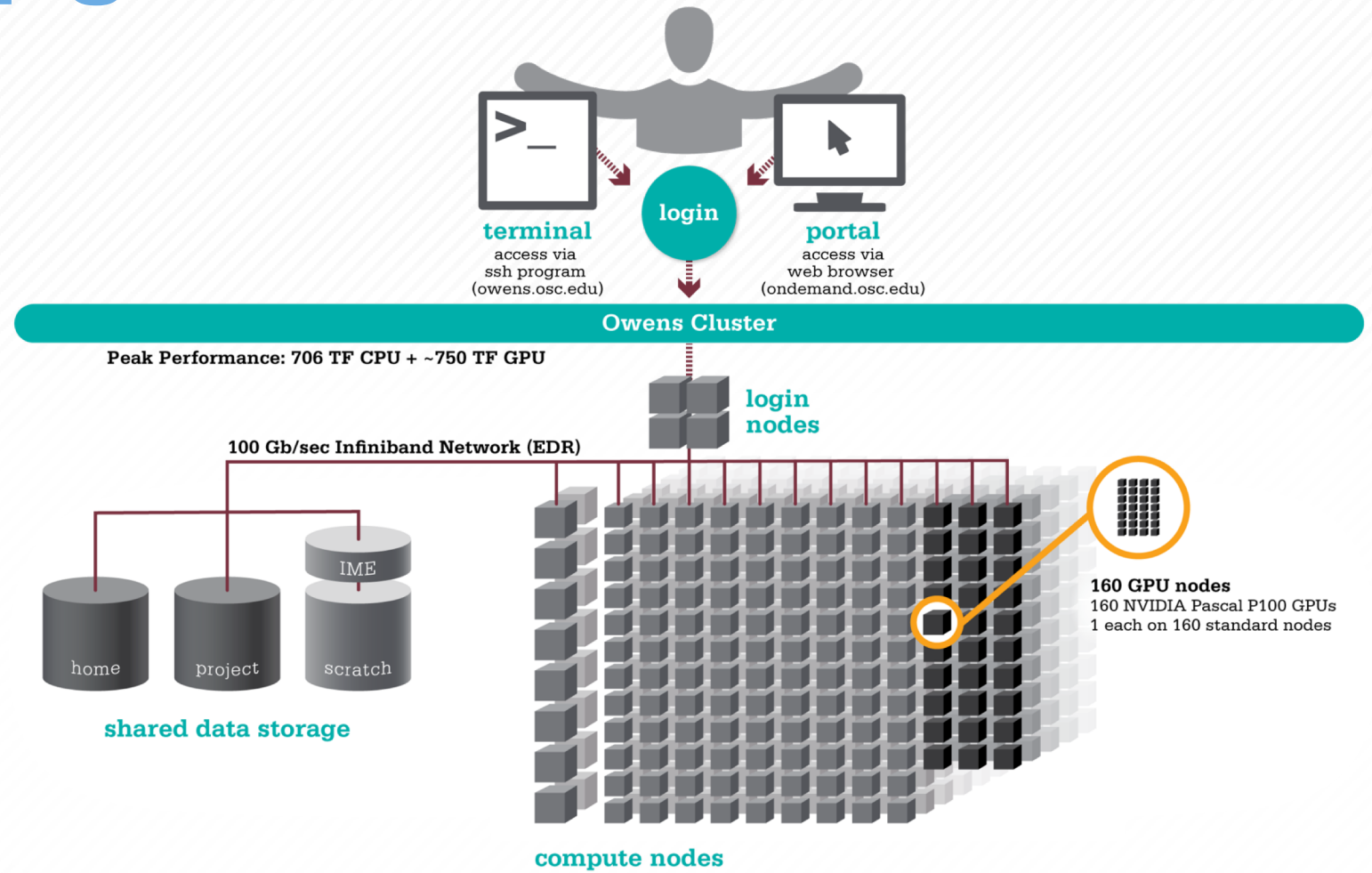
	Pitzer (2018)	Owens (2016)	Ruby (2014)
Theoretical Performance	~1300 TF	~1600 TF	~144 TF
# Nodes	260	824	240
# CPU Cores	10,560	23,392	4,800
Total Memory	~70.6 TB	~120 TB	~15.3 TB
Memory per Core	>5 GB	>5 GB	3.2 GB
Interconnect	EDR IB	EDR IB	FDR/EN IB



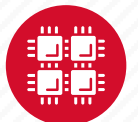
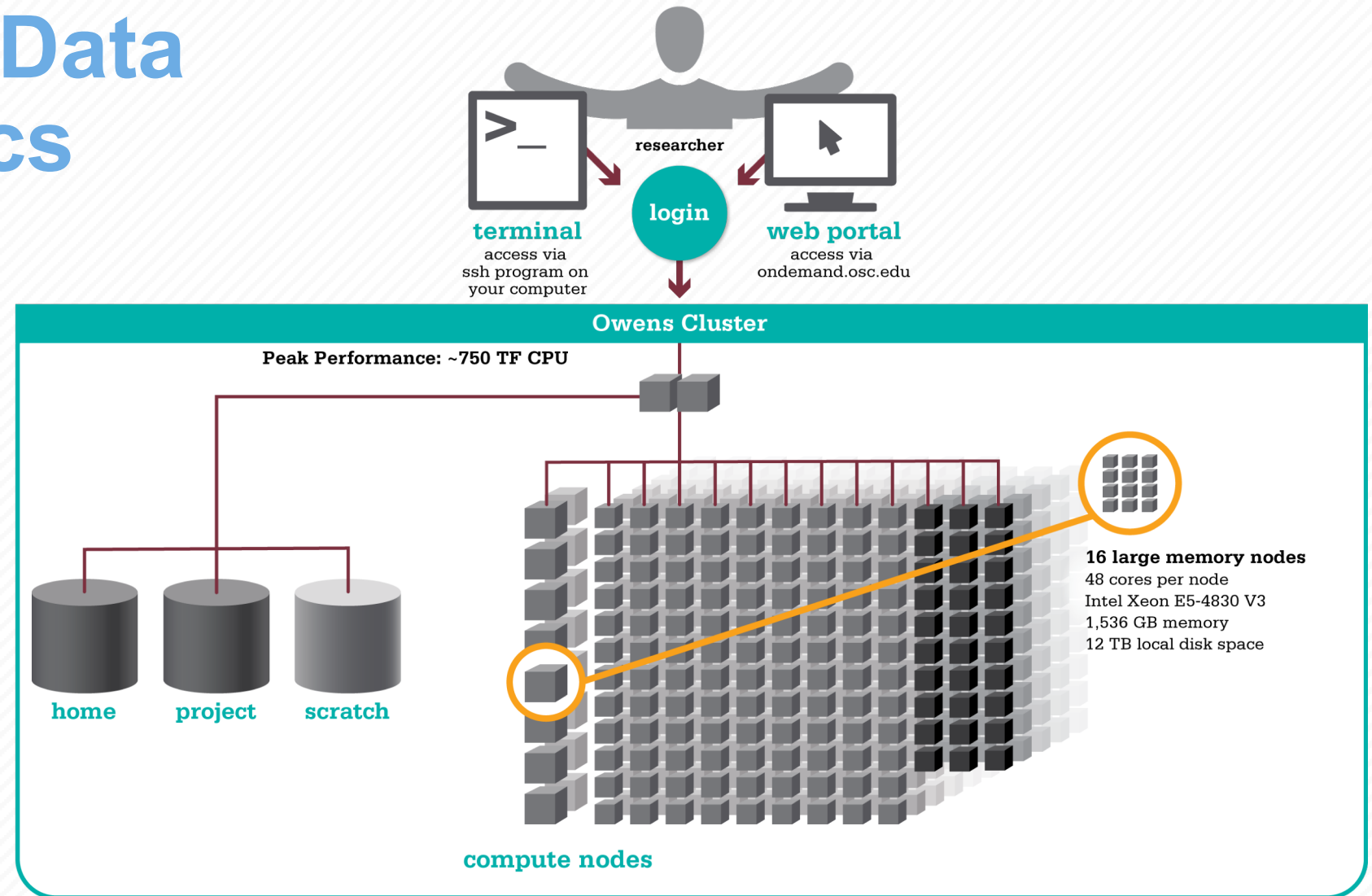
Owens Compute Nodes



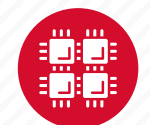
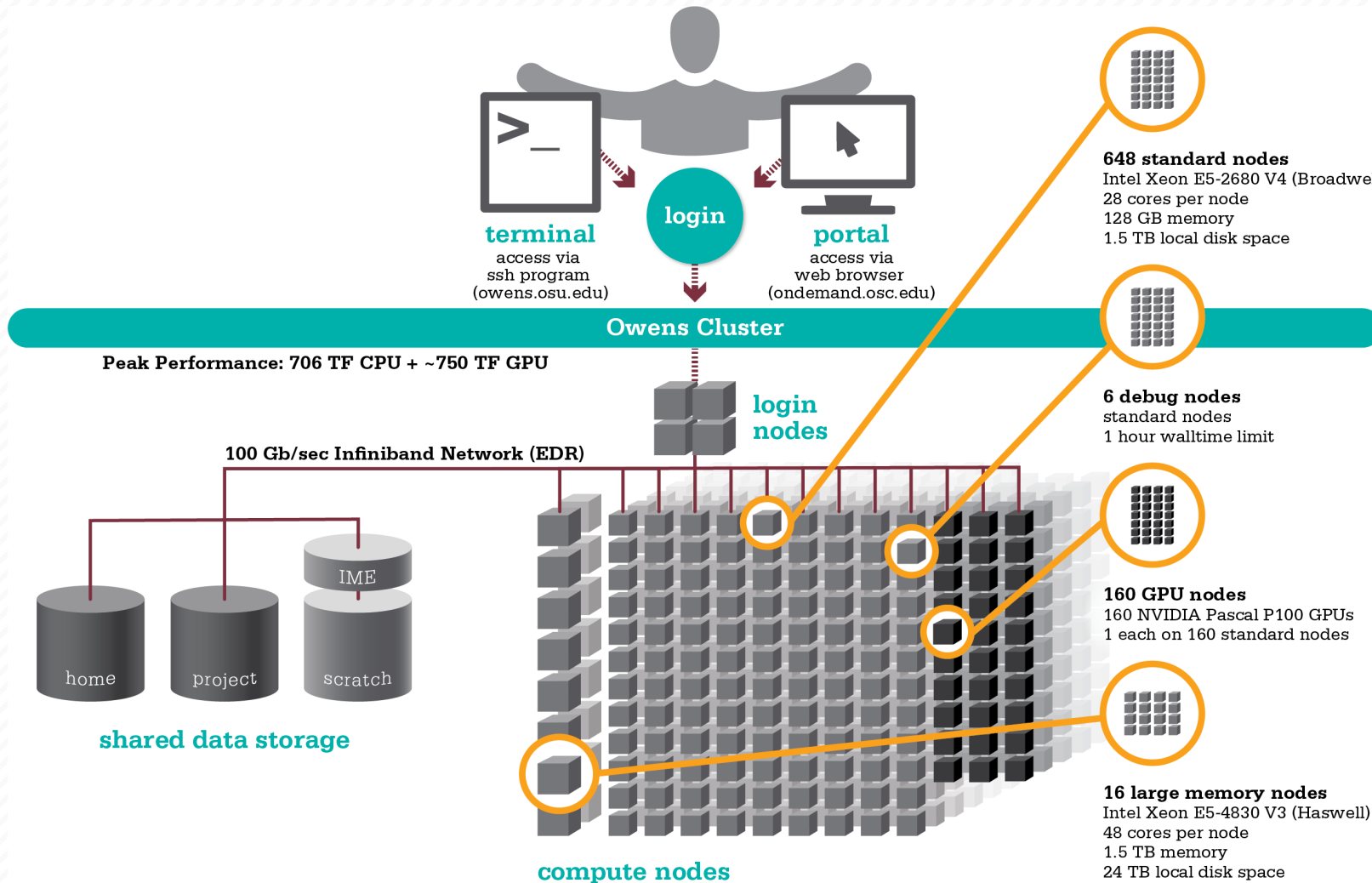
Owens GPU Nodes



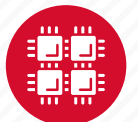
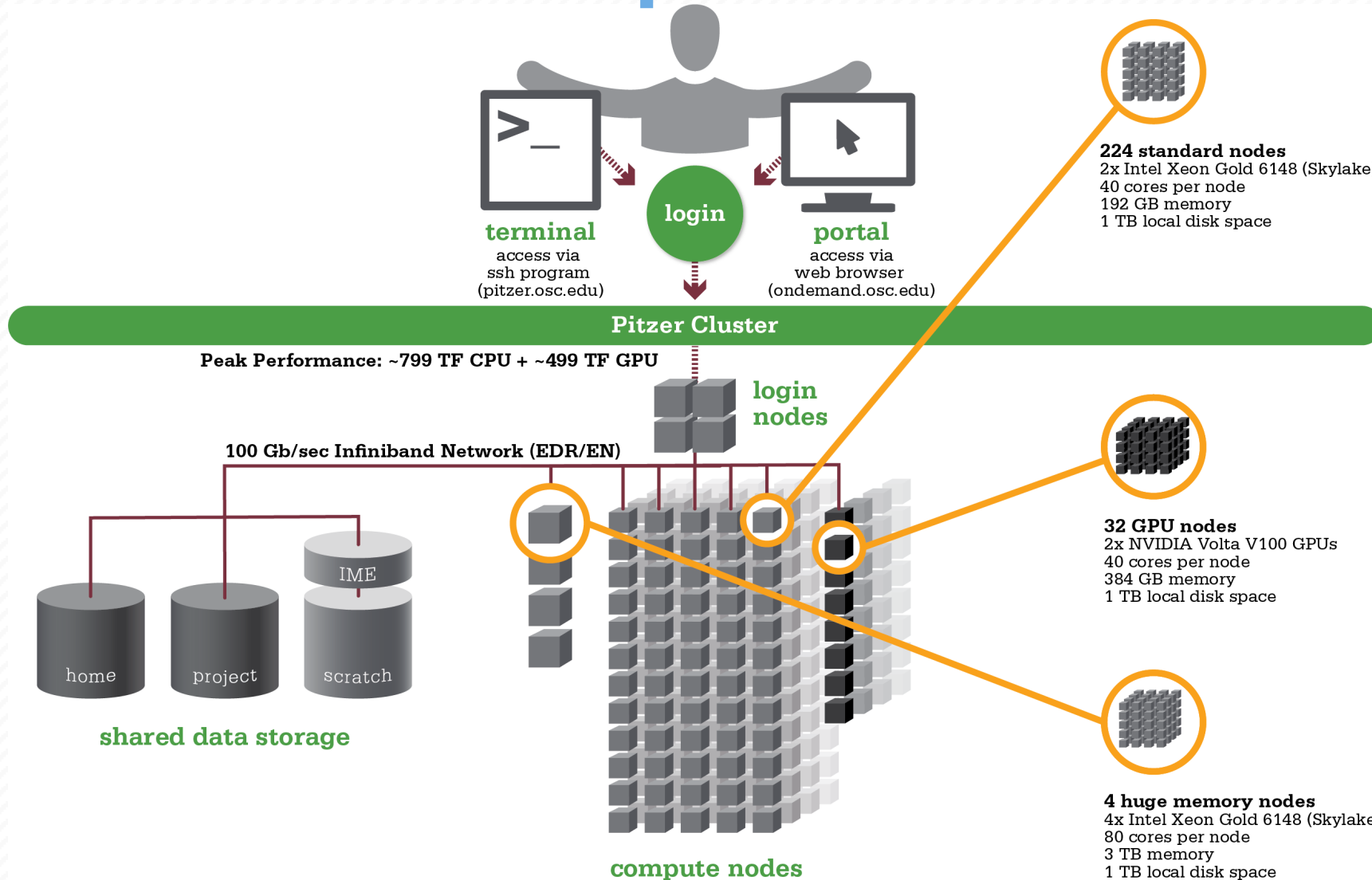
Owens Data Analytics Nodes



Owens Cluster Specifications

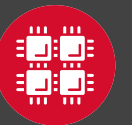


Pitzer Cluster Specifications



Login Nodes – Usage

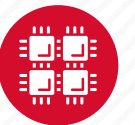
- Purpose
 - Submit jobs to batch system
 - Edit files
 - Manage your files
 - Interactive work – small scale
- Limits
 - 20 minutes CPU time
 - 1GB memory
- **Use the batch system for serious computing!**



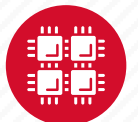
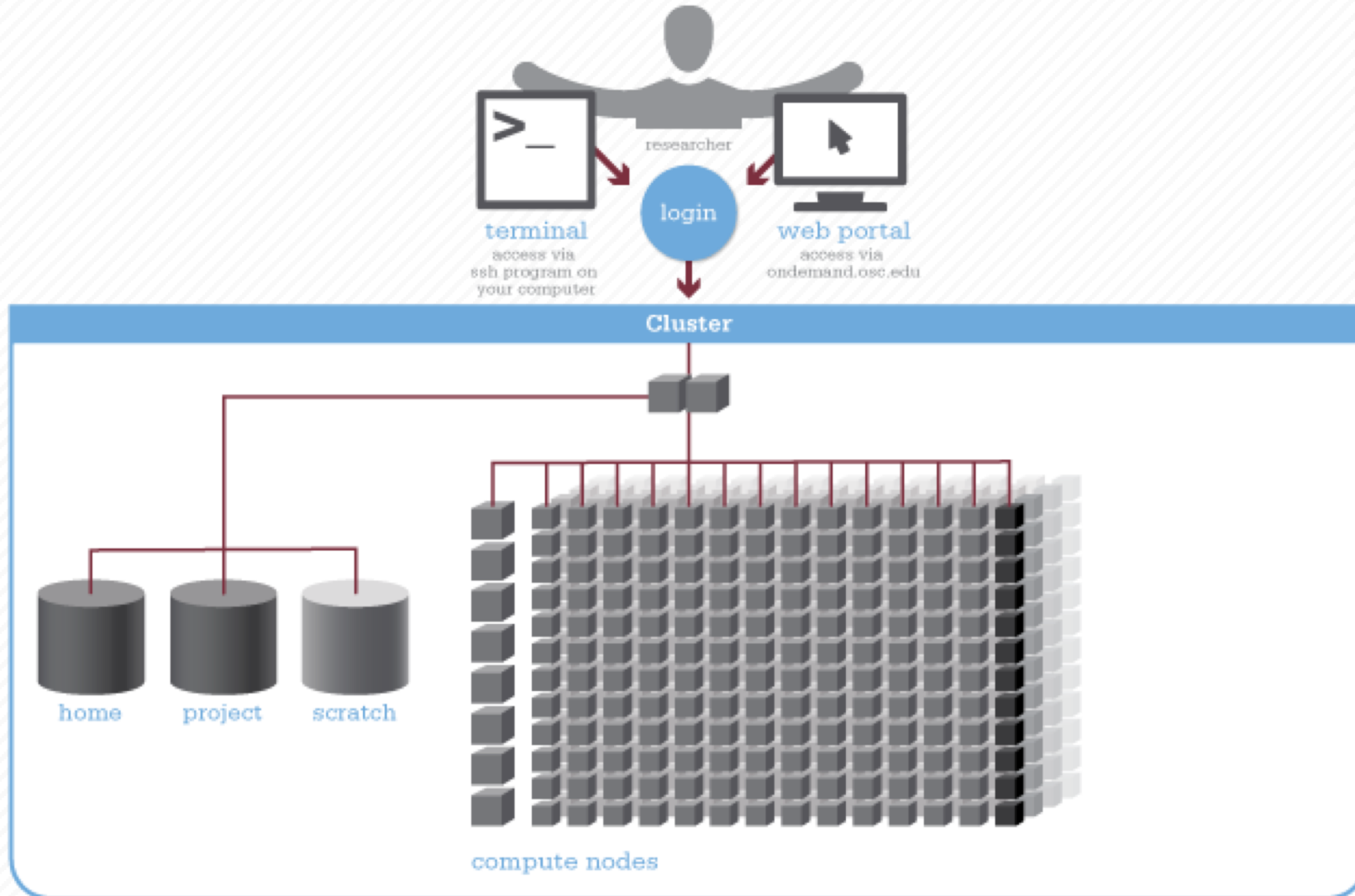


Data Storage Systems

"War is ninety percent information." – Napoleon Bonaparte

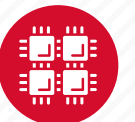


Four different file systems

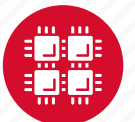


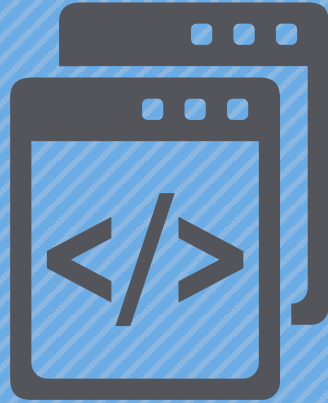
Filesystem Overview

- 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



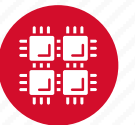
Filesystem	Quota	Backed-Up?	Purged?
Home (\$HOME)	500GB	Yes	No
Project (/fs/project)	By request	Yes	No
Scratch (/fs/scratch)	None	No	Yes – 120 days
Compute (\$TMPDIR)	800GB (Oakley), 1 TB (Ruby & Owens)	No	Yes – when job completes





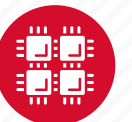
Getting Started at OSC

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



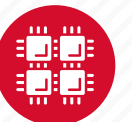
Who can get an OSC project?

- 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



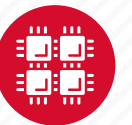
Accounts and Projects at OSC

- 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



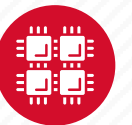
Allocations and Charges

- 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

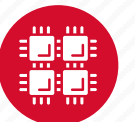
- The first 10,000 RUs on all academic projects are subsidized
- Usage above 10,000 RUs will be billed to the PI's institution at \$0.075/RU
- PIs should contact their Office of Research for details on local billing



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



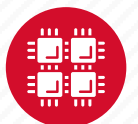
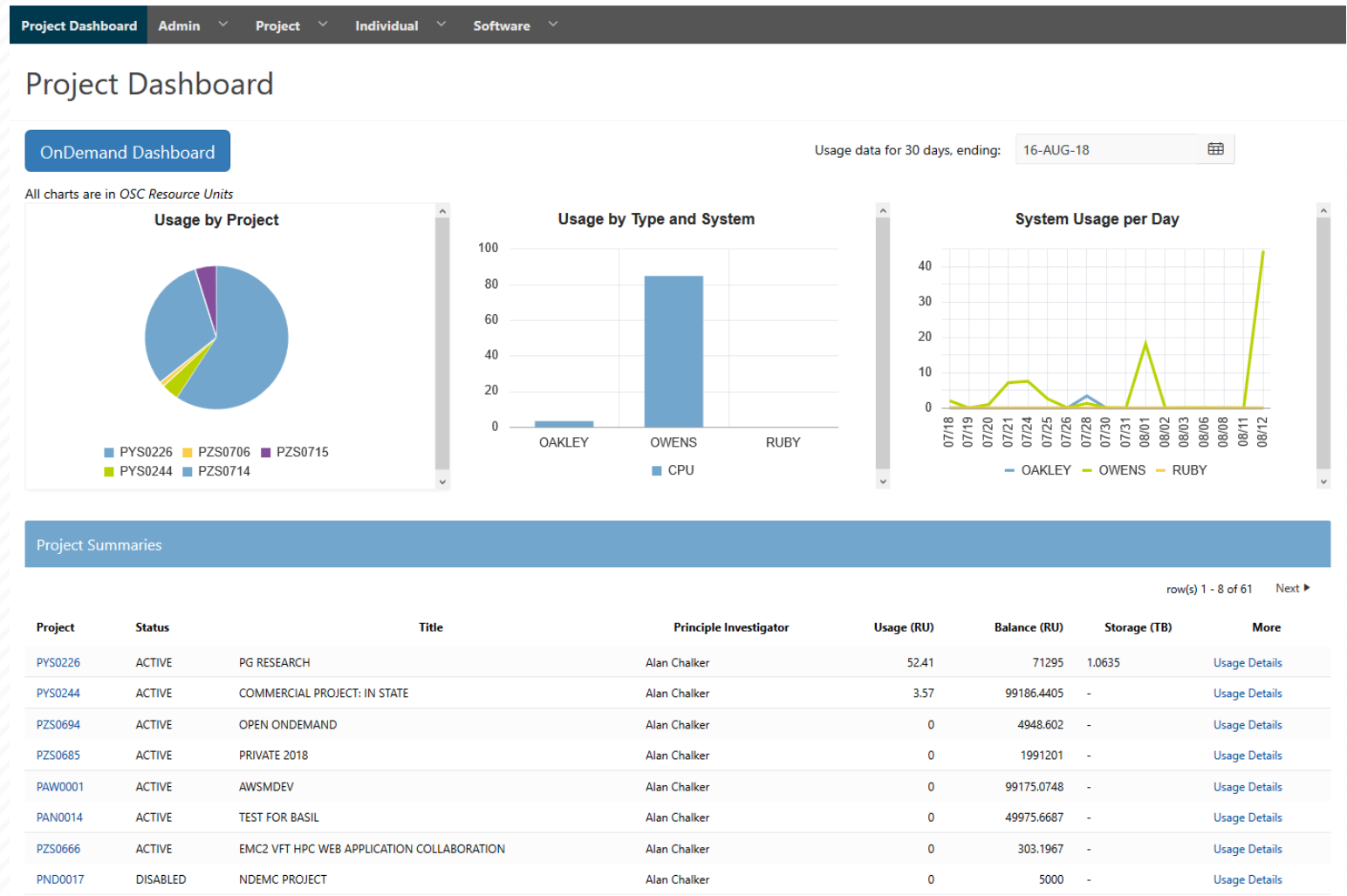
Client Portal– my.osc.edu

Features

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

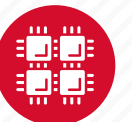
PI resources

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



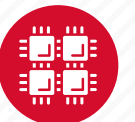
Your Contact Info

- 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



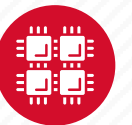
Statewide Users Group (SUG)

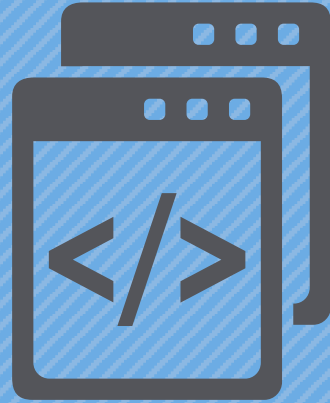
- The Statewide Users Group (SUG) is made up of OSC users
 - Provides program and policy advice to OSC
 - Meets twice a year
 - Headed by a chairperson elected yearly
- Standing committees
 - Allocations
 - Software and Activities
 - Hardware and Operations
- Get involved!
 - Next meeting will be April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



Citing OSC

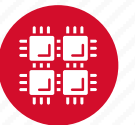
- Please cite OSC in your publications:
 - Details at www.osc.edu/citation
- These publications should be reported to OSC





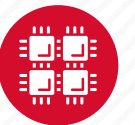
User Environment

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



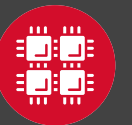
Linux Operating System

- “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

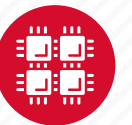
- Connect to OSC machines using `ssh` (secure shell)
 - From a Linux/UNIX machine : At prompt, enter
 - `ssh userid@owens.osc.edu`
 - `ssh userid@ruby.osc.edu`
 - From a Mac: Enter `ssh` command in TERMINAL window
 - From Windows: `ssh` client software needed
 - Both commercial and free versions are 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

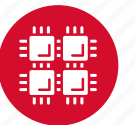
- Web based
 - Usable from computers, tablets, smartphones
 - Zero installation
- Single point of entry
 - User needs three things
 - ondemand.osc.edu
 - OSC Username
 - OSC Password

Connected to all resources
at OSC

- 2: Interactive Services

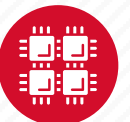
- File Access
- Job Management
- Visualization Apps
 - Desktop access
 - Single-click apps (Abaqus, Ansys, Comsol, Paraview)
- Terminal Access

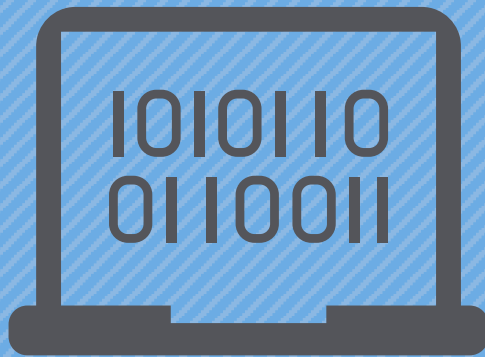
Tutorial available at
osc.edu/ondemand



Transferring Files to and from the Cluster

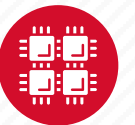
- Most file transfers to and from OSC machines use **sftp** or **scp**
 - Linux and Mac have them built in
 - Windows needs extra software - FileZilla
- For small files, connect to a login node
owens.osc.edu
- For large files, transfer may fail due to shell limits
 - Connect to **sftp.osc.edu** (file transfer only)
- OnDemand drag and drop file transfer up to 5GB files



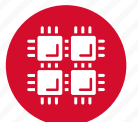
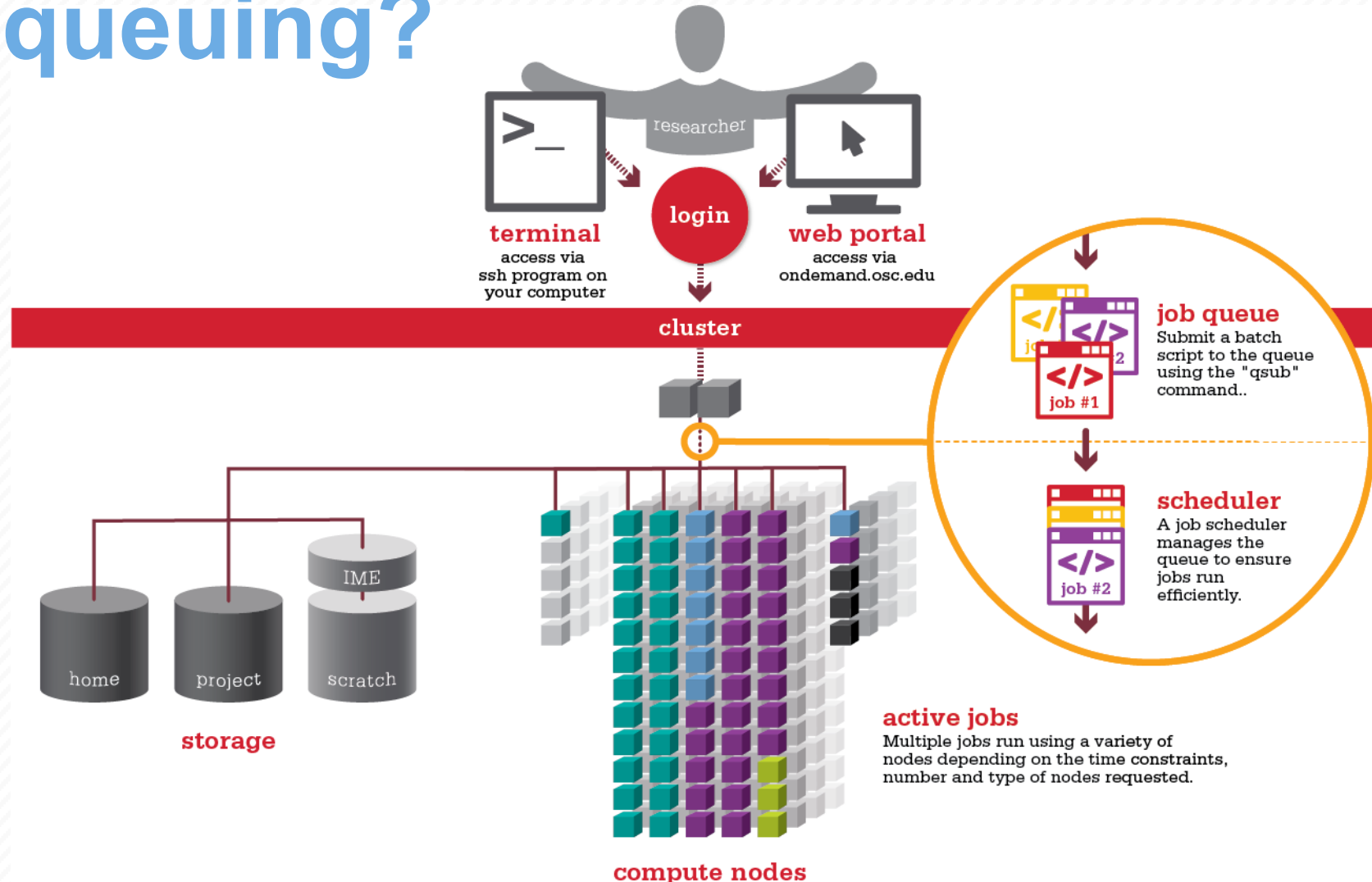


Batch Processing

“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

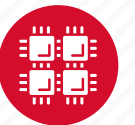


Why do supercomputers use queuing?



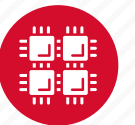
Batch System at OSC

- 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



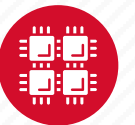
Idea Behind Batch Processing

- 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



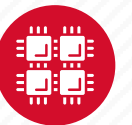
Steps for Running a Job on the Compute Nodes

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

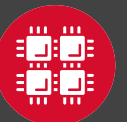
```
#PBS -N serial_fluent
#PBS -l walltime=1:00:00
#PBS -l nodes=1:ppn=28:gpus=1
#PBS -A PAS####
#PBS -j oe
#PBS -l software=fluent+1
# Set up the FLUENT environment
module load fluent
# Move to directory job was submitted from
cd $PBS_O_WORKDIR
# Copy input files to compute node
cp run.input $TMPDIR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

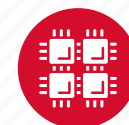
Put all this into a text file!



Submitting a Job and Checking Status

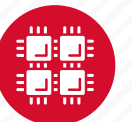
- Command to submit a job
 - `qsub script_file`
- Response from PBS (example)
 - `123456.owens-batch.ten.osc.edu`
- Show status of batch jobs
 - `qstat -a jobid`
 - `qstat -u username`
 - `qstat -f jobid`
- Delete a job
 - `qdel jobid`

[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



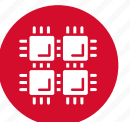
Job Output

- 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
```

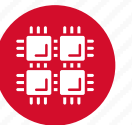
```
-----  
Resources used:  
cput=125:18:32  
walltime=02:14:32  
mem=34.824GB  
vmem=77.969GB
```

```
-----  
Resource units charged (estimate):  
12.556 RUs  
-----
```



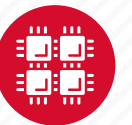
Interactive Batch Jobs

- 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 -l nodes=2:ppn=28 -l walltime=1:00:00 -m abe`



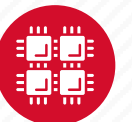
Batch Queues

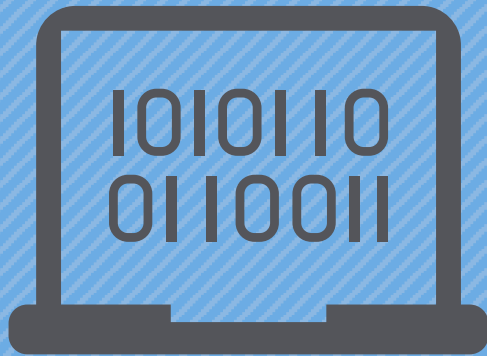
- The three clusters have separate batch systems
 - Submit job and check status on the same cluster
- Debug reservation
 - A few nodes on each system are reserved for short jobs (≤ 1 hour)
 - Special flag required: `-q debug`



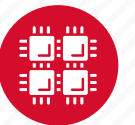
To Take Advantage of Parallel Computing

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



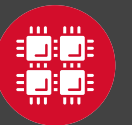


Loading and Running Software



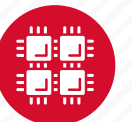
Software Maintained by OSC

- 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



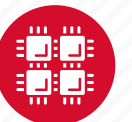
Modules for Software access

- 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`



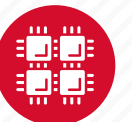
Adding or Removing Software from Your Environment

- 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



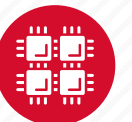
Module Commands

- 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`



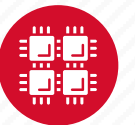
Third party applications

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



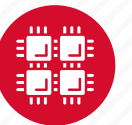
Third party applications

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



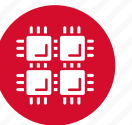
Access to Licensed Software

- 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

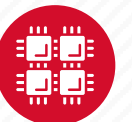
Tutorial materials: https://khill42.github.io/OSC_IntroHPC/

Office Hours:

go.osu.edu/rc-osc Tuesdays 1-3 p.m. or Weekdays 4-5 at Pomerene Hall

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

 twitter.com/osc

 facebook.com/ohiosupercomputercenter

 osc.edu

 oh-tech.org/blog

 linkedin.com/company/ohio-supercomputer-center