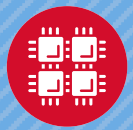
Ohio Supercomputer Center

An OH·TECH Consortium Member

An introduction to OSC services, hardware, and environment

September 15, 2020



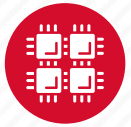
Kate Cahill

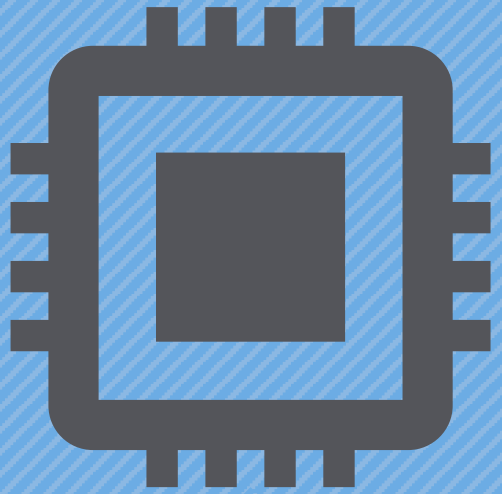
Education & Training Specialist

“OSC is here to empower your research.”

Outline

- What is OSC?
- High-Performance Computing (HPC) Concepts
- Hardware Overview
- Getting a New Project/Account
- User Environment
- Using Software on OSC systems
- Batch Processing
- OnDemand demo



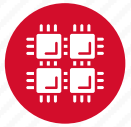


What is the Ohio Supercomputer Center?

About OSC

- Founded in 1987, through the Ohio Department of Higher Education

- Statewide resource for all universities in Ohio
  - high performance computing services
  - computational science expertise



Service Catalog

Cluster Computing

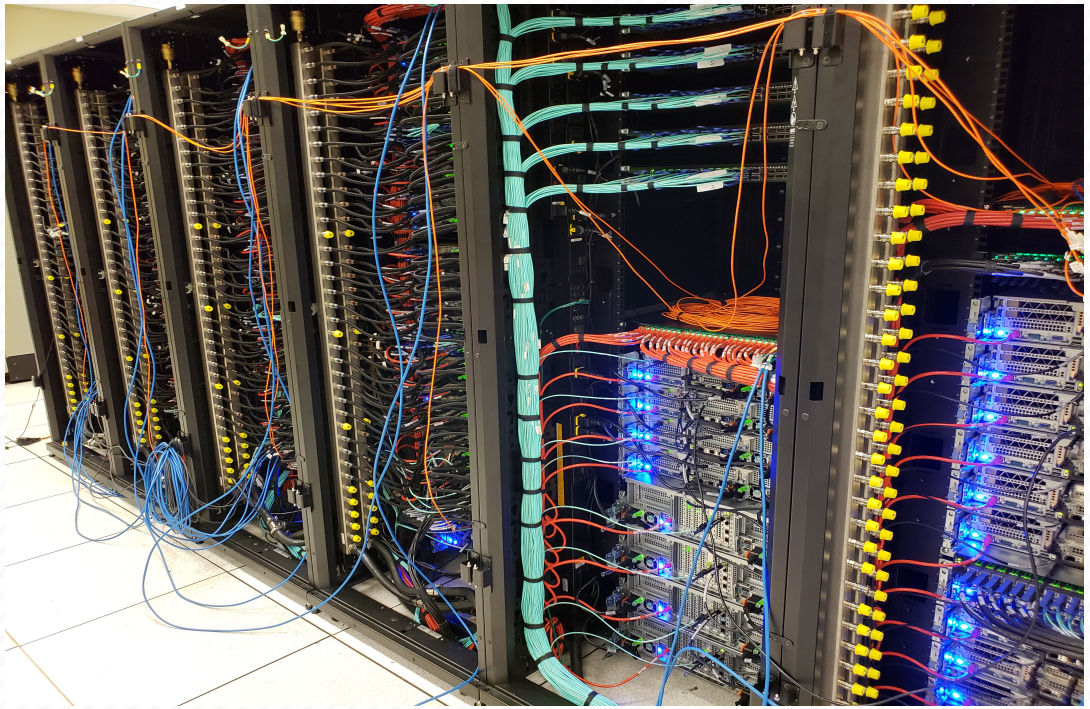
A fully scalable center with mid-range machines to match those found at National Science Foundation centers and other national labs.

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.

Client Services

CY2019

28 Ohio universities

49 companies

49 universities outside of Ohio

4,246 clients

59 college courses used OSC

292 new projects created

757 projects served

25 training opportunities

354 trainees

228 publications cited OSC

OSC Classroom Usage

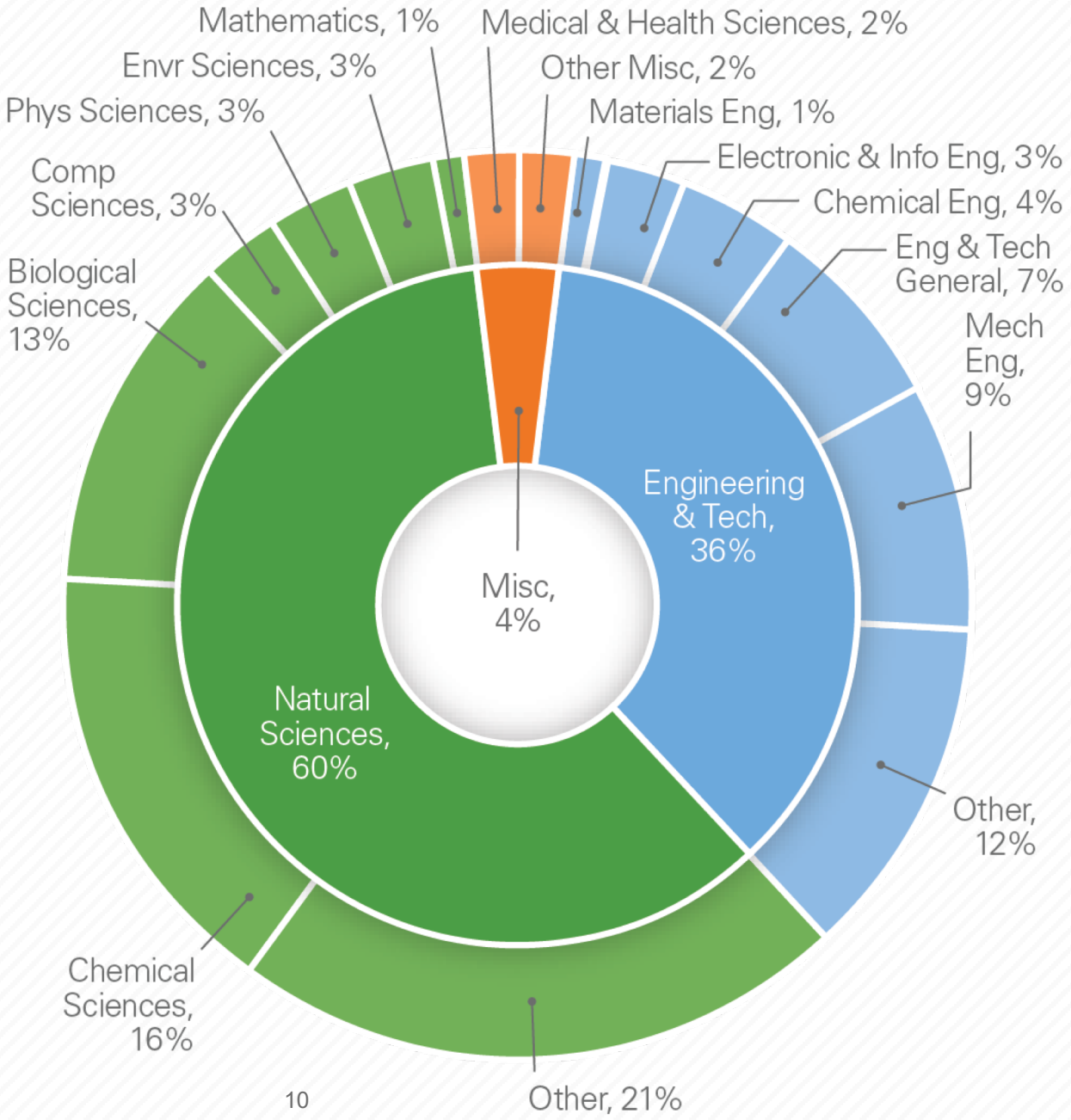
CY 2019

1,829 students, 59 departments, 13 universities

Institution	# of Students	# of Courses	# of Departments
Bowling Green State University	10	2	2
Cleveland State University	11	1	1
Kent State University	18	3	3
Miami University	46	2	2
Mount Union College	4	1	1
Ohio State University	1,354	36	36
Ohio University	10	1	1
Stark State College	9	1	1
University of Akron	9	1	1
University of Cincinnati	313	8	8
University of Toledo	3	1	1
Wittenberg University	8	1	1
Wright State University	34	1	1

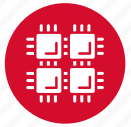
Usage by Field of Science*

CY2019



*Fields of science are self-reported and classified based on "Revised Field of Science and Technology (FOS) Classification in The Frascati Manual" found here https://www.oecd.org/science/inno/38235147.pdf

HPC Concepts



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

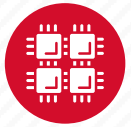
What is the difference between your laptop and a supercomputer?

X 10000 =

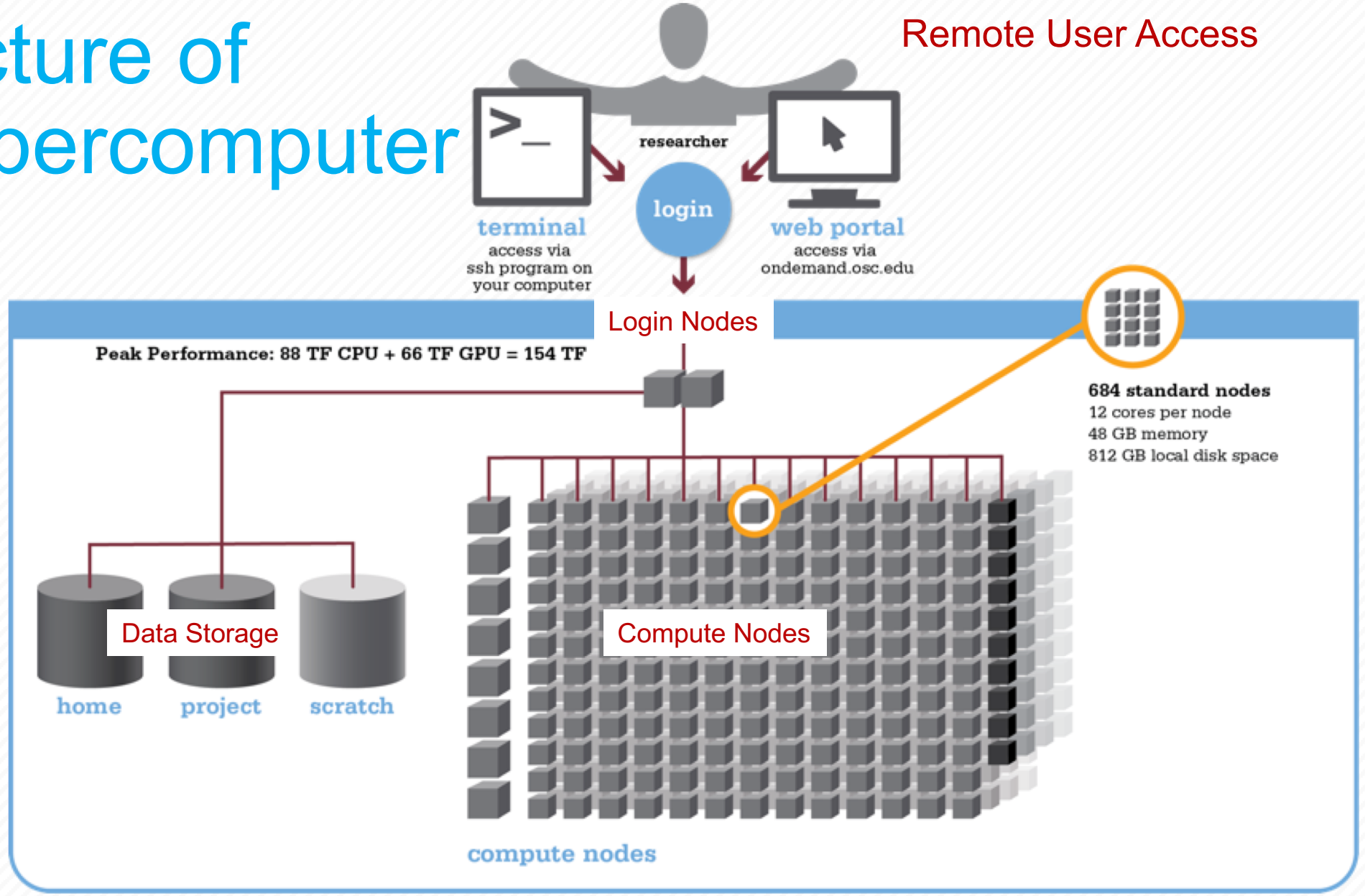
Remote Connect

HPC Terminology

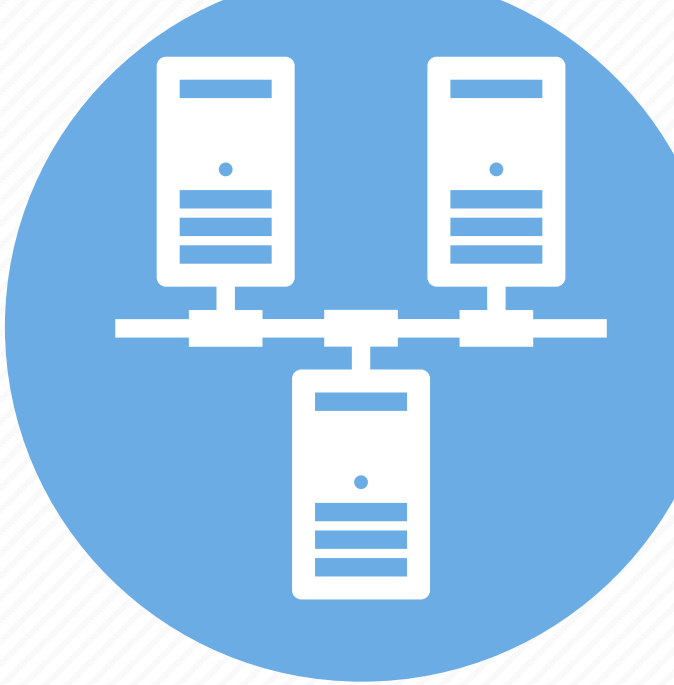
- Cluster
  - A group of computers (nodes) connected by a high-speed network, forming a supercomputer
- Node
  - Equivalent to a high-end workstation, part of a cluster
- Graphical Processing Unit (GPU)
  - A separate multi-core processor that can handle many small calculations
- Core
  - A processor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



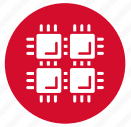
Structure of a Supercomputer



Memory



- Holds data that is being calculated on, as well as computational instructions
- *Shared memory* is local to one node and several process
- *Distributed memory* is on multiple nodes and each process normally has its own copy or part of the data.



Storage

Longer term location for data not currently in use

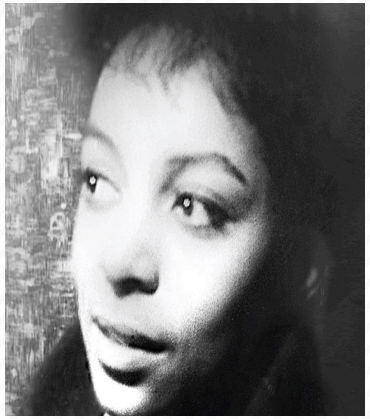
Different types of “disk” for different needs

- Local disk in the node, often SSD
- Shared scratch
  - Short-term storage only!!
- Long-term or archive

Hardware Overview

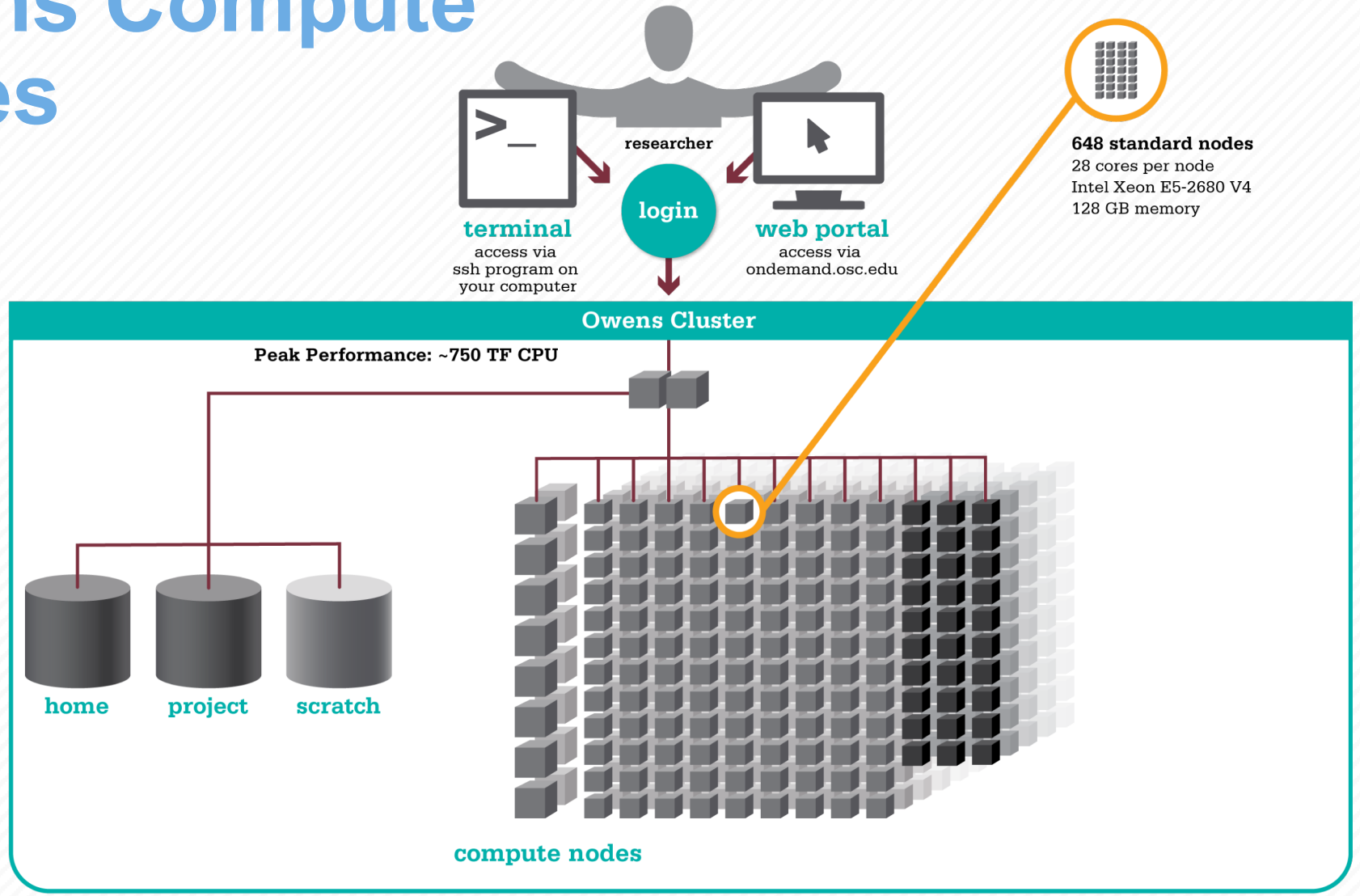
System Configurations

	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

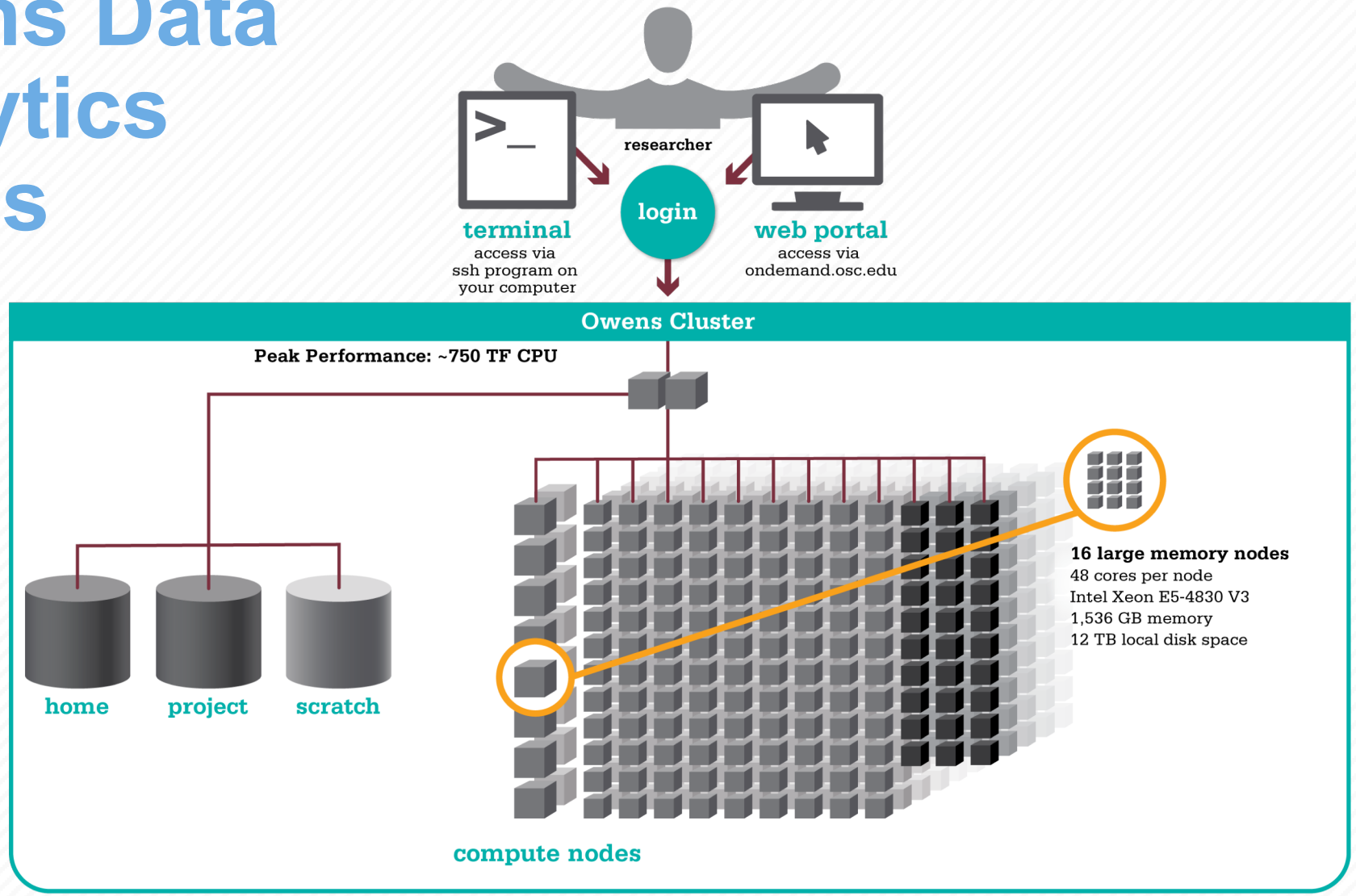


*Pitzer expansion coming Fall 2020

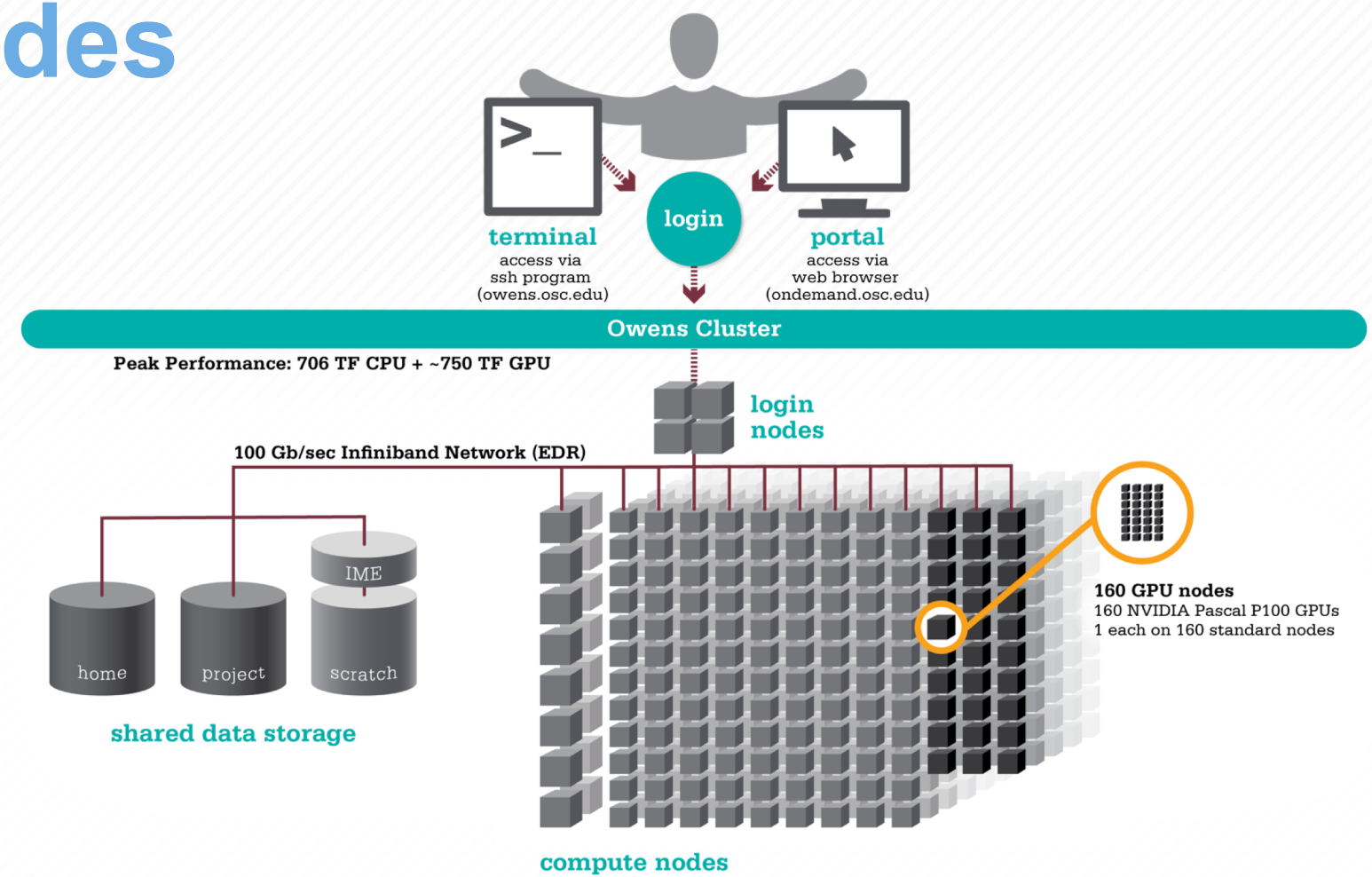
Owens Compute Nodes



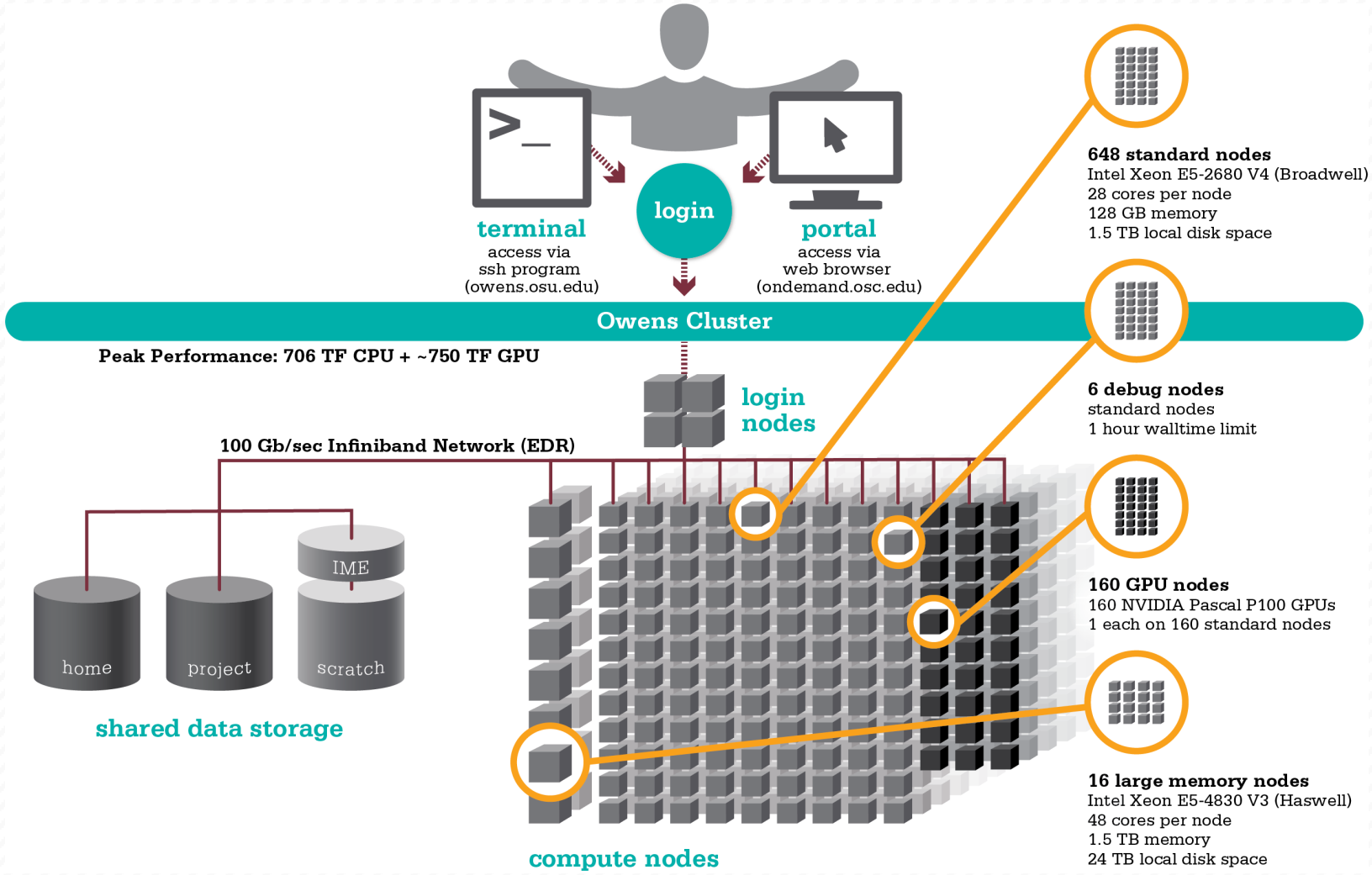
Owens Data Analytics Nodes



Owens GPU Nodes

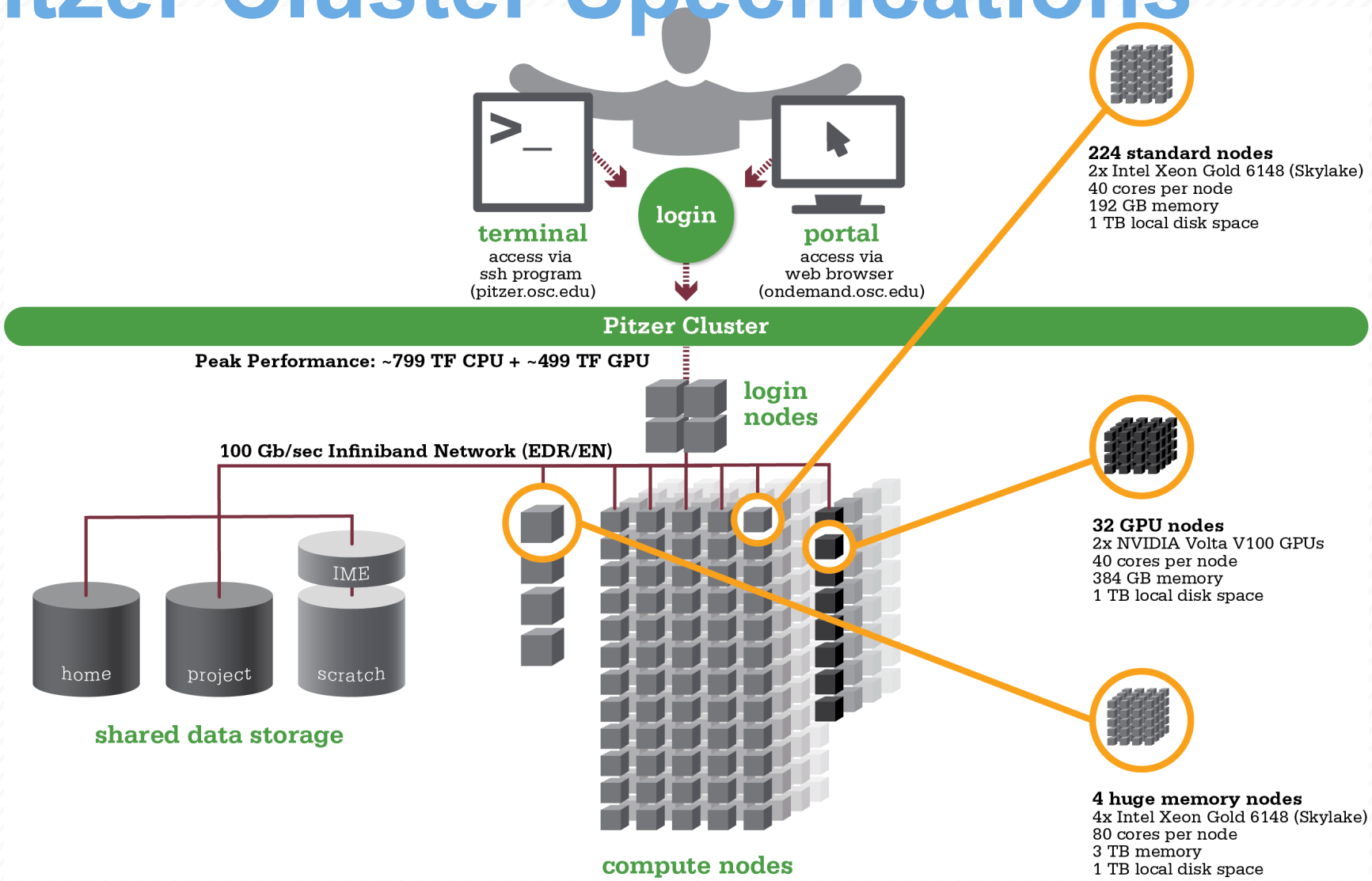


Owens Cluster Specifications



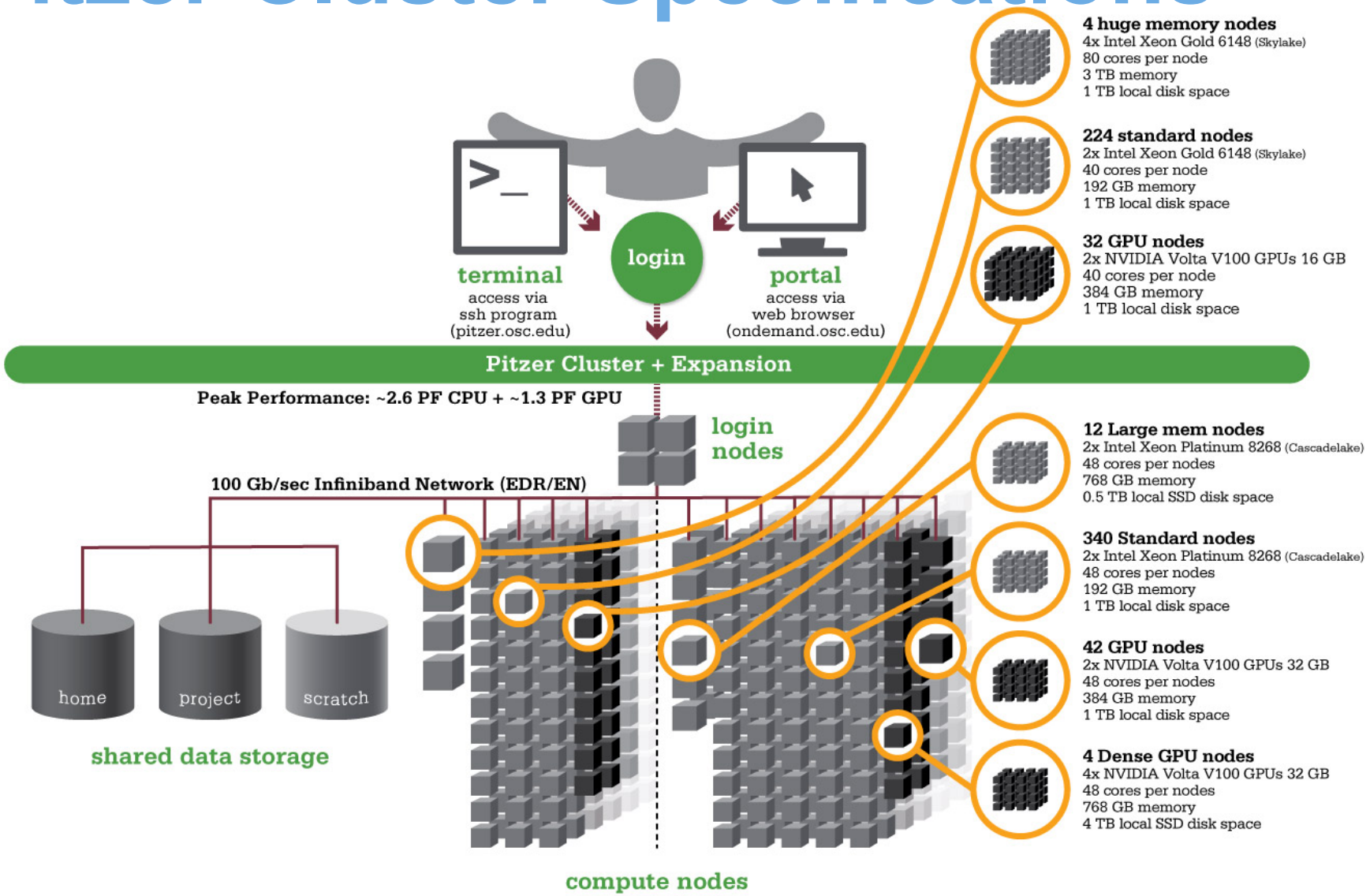
https://www.osc.edu/resources/technical_support/supercomputers/owens

Pitzer Cluster Specifications



https://www.osc.edu/resources/technical_support/supercomputers/pitzer

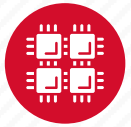
Pitzer Cluster Specifications



https://www.osc.edu/resources/technical_support/supercomputers/pitzer

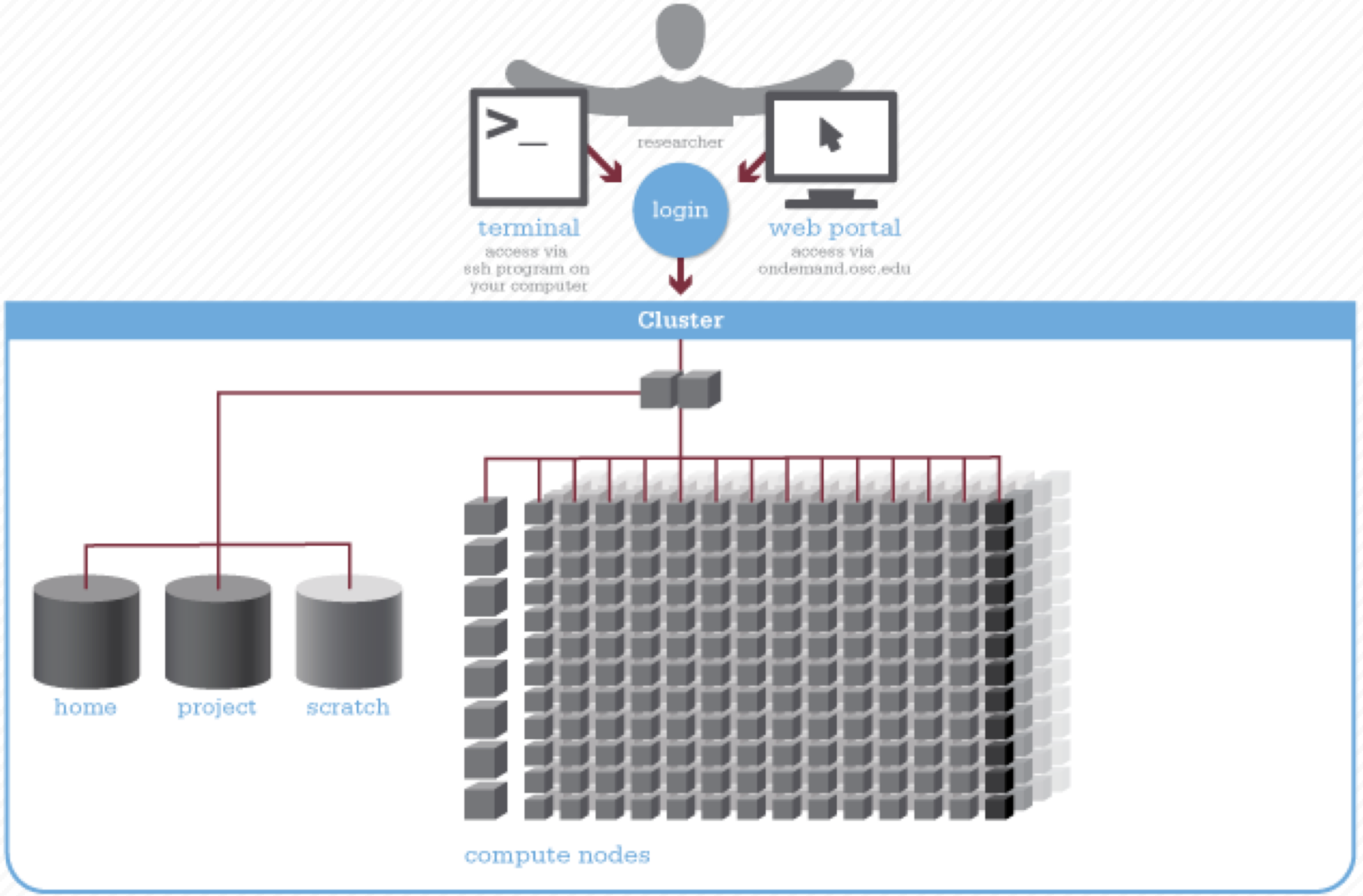
Login Nodes – Usage

- 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

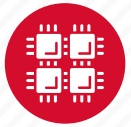
Four different file systems



Filesystem Overview

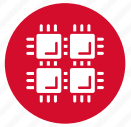
- Home
  - Store your files here, backed up daily
  - Use $HOME or ~*username* to reference location
- Project/ESS
  - Available to Project PIs by request; shared by all users on a project, backed up daily
  - Use /fs/project/*project#* or /fs/ess/*project#* to reference location
- Scratch
  - Store 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

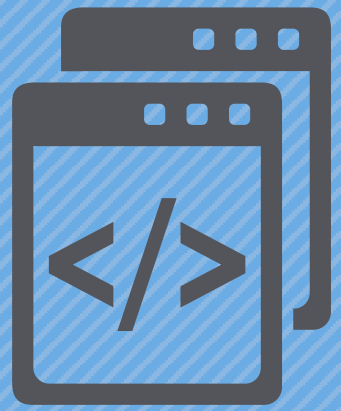
https://www.osc.edu/supercomputing/storage-environment-at-osc/available-file-systems



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

https://www.osc.edu/supercomputing/storage-environment-at-osc/available-file-systems

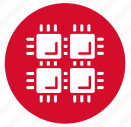




Getting Started at OSC

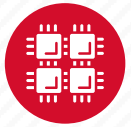
Who can get an OSC project?

- 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
- Commercial projects
  - Commercial organizations may purchase time on OSC systems



Accounts and Projects at OSC

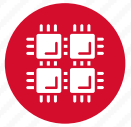
- Project
  - Headed by a PI
  - May include other users
  - Oversees computing resources for a project
- Account
  - Username and password to access HPC systems
  - Each account used by one person
  - If you work on multiple projects, you will have one account that can access all of them



Usage Charges

- Charges are in terms of core hours, GPU hours, TB months
- Project has a dollar balance
- Services, e.g. compute and storage, are charged to a project
- General Compute, GPU, Huge Memory, Storage costs are still partially subsidized and highly competitive

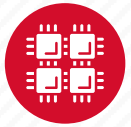
https://www.osc.edu/content/academic_fee_model_faq



Ohio Academic Projects

- Standard Projects
  - Each PI can receive $1,000 grant annually to cover OSC services
  - PI can set a budget so no unexpected charges
  - No more proposal submissions
- Classroom projects are fully subsidized
- Request at my.osc.edu

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



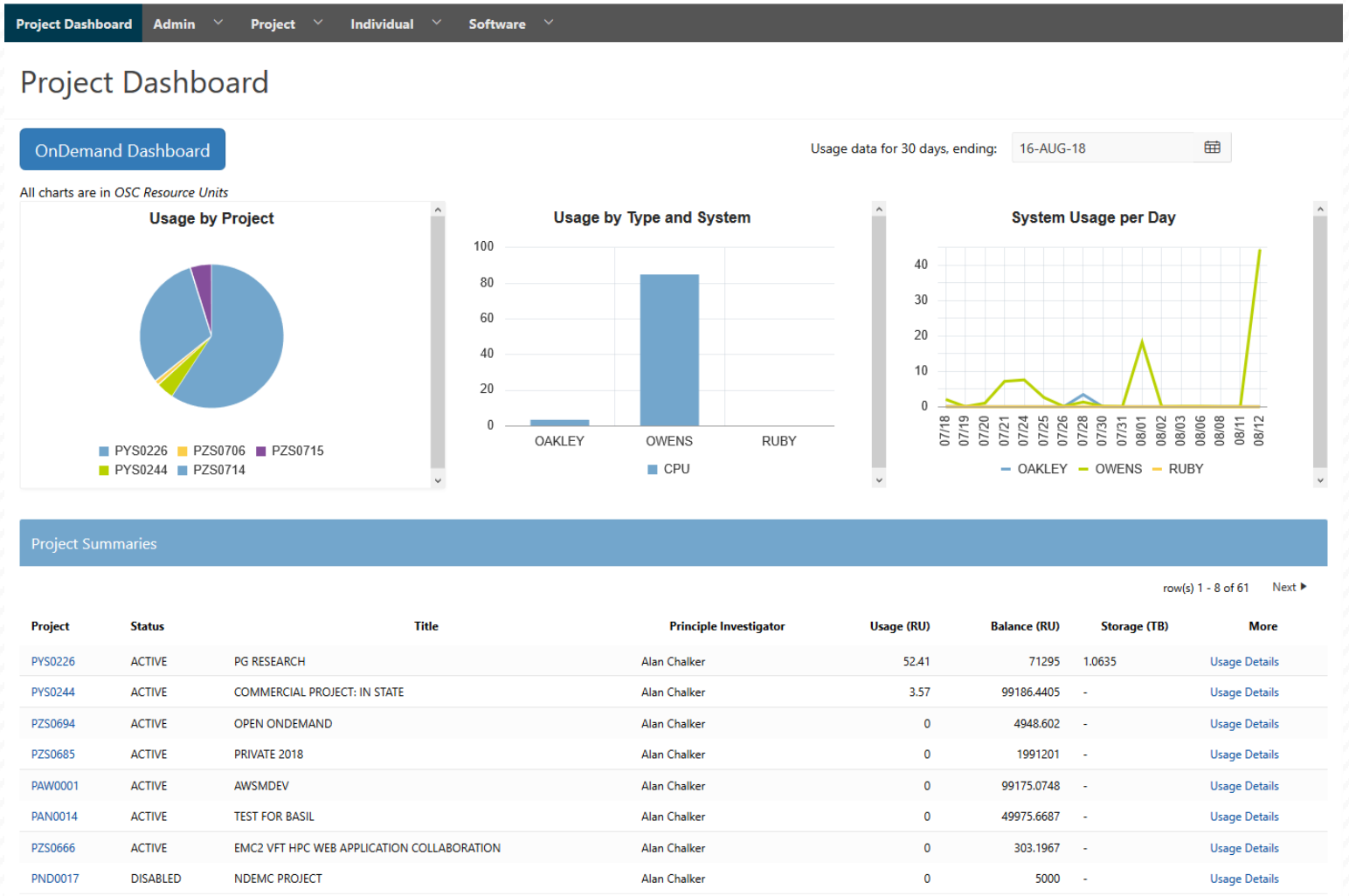
Client Portal– my.osc.edu

Features

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

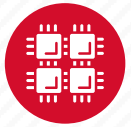
PI resources

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



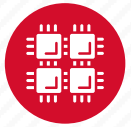
Statewide Users Group (SUG)

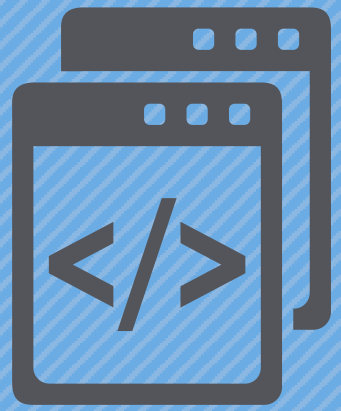
- The Statewide Users Group (SUG) is made up of OSC users
  - Provides program and policy advice to OSC
  - Meets twice a year
  - Headed by a chairperson elected yearly
- Standing committees
  - Allocations
  - Software and Activities
  - Hardware and Operations
- Get involved!
  - Virtual SUG October 22!



Citing OSC

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





User Environment

Linux Operating System

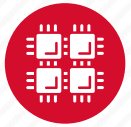
- “UNIX-like”
- Widely used in HPC
- Mostly command-line
- Choice of shells (bash is default)
- Freely distributable, open-source software
- Tutorials available
- www.linux.org

Connecting to an OSC Cluster

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

    `ssh *userid*@owens.osc.edu`
  - From Windows: **`ssh`** client software needed
    - Both commercial and free versions are available
- Connect using OSC OnDemand portal (web-based)
- Connect with graphics. Programs can have an X-based GUI
  - Linux/UNIX and Mac: Use **`-X`** flag

    `ssh -X *userid*@owens.osc.edu`
  - Windows: extra software needed for X11 forwarding
  - Programs run primarily on log in nodes. Can also submit batch job



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

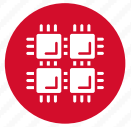
osc.edu/ondemand

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
- GLOBUS-- a large life transfer system

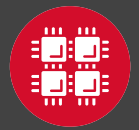
https://www.osc.edu/resources/getting_started/howto/howto_use_globus_overview



Using and Running Software at OSC

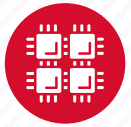
Software Maintained by OSC

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



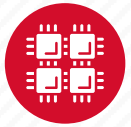
Third party applications

- **General programming software (**$statewide licensed)
  - gnu compilers and debugger
  - $ Intel compilers
  - $ Arm DDT debugger
  - $ Arm MAP profiler
  - $ ANSYS
  - MPI library
  - HDF5
  - NetCDF
  - Java, Java Virtual Machine
  - Python
  - R Statistical & Programing environment



Third party applications

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

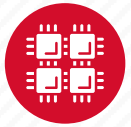


Access to Licensed Software

- 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:

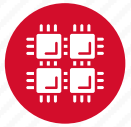
  http://www.osc.edu/supercomputing/software/



OSC doesn't have the software you need?

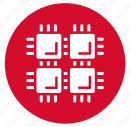
- Commercial software
  - Fill out a request form (see our FAQ)
  - SUG will consider it
- Open-source software
  - You can install it yourself in your home directory*
  - If there's enough demand, we can install it for shared use
- Have your own license?
  - Contact OSC Help

*https://www.osc.edu/resources/getting_started/howto/howto_locally_installing_software



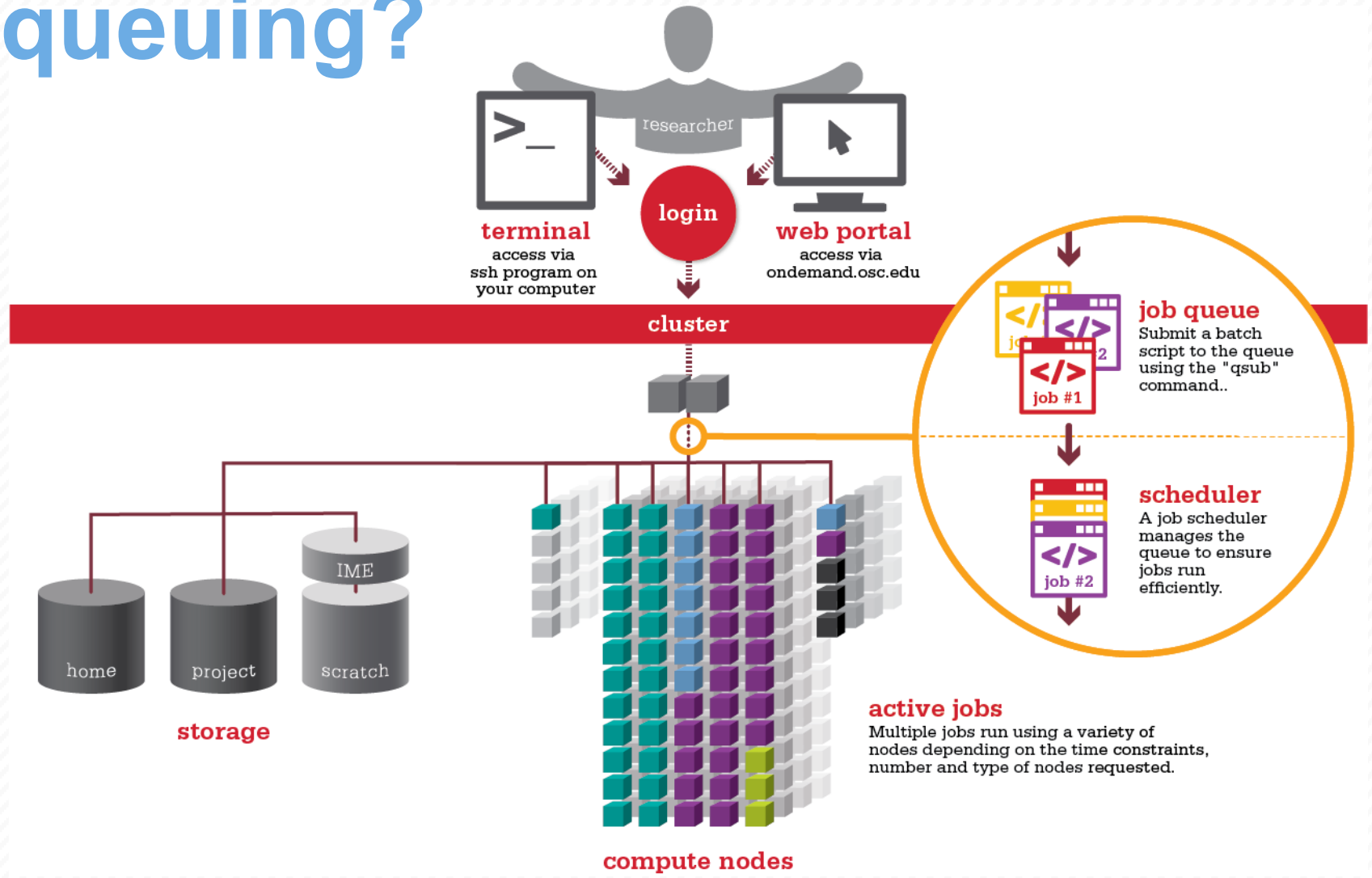
Loading and Running Software

- 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`



Batch Processing

Why do supercomputers use queuing?

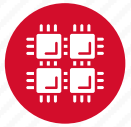


Steps for Running a Job on the Compute Nodes

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

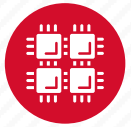
- Nodes and cores (processors) per node, GPUs
- Memory (optional)
- Walltime
  - Overestimate slightly – job will be deleted if it hits limit
  - Shorter job may start sooner due to backfill
- Project #
- Software licenses
  - See specific software page on OSC website



Batch changes at OSC

- OSC Clusters will switch from Torque/Moab to SLURM by Dec. 31
  - Pitzer Sept 22
  - Owens later this fall
- Compatibility layer active so most PBS scripts still work
- New users – SLURM
- Current users – test scripts or use SLURM

https://slurm.schedmd.com/



Sample Torque/Moab Batch Script

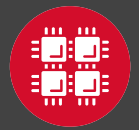
```
#PBS -N serial_fluent
#PBS -l walltime=1:00:00
#PBS -l nodes=1:ppn=28:gpus=1
#PBS -A PAS####
#PBS -j oe
#PBS -l software=fluent+1
# Set up the FLUENT environment
module load fluent
# Move to directory job was submitted from
cd $PBS_O_WORKDIR
# Copy input files to compute node
cp run.input $TMPDIR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



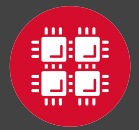
Sample SLURM Batch Script

```
#!/bin/bash
#SBATCH --time=1:00:00
#SBATCH --nodes=2 --ntasks-per-node=40
#SBATCH --job-name=hello
#SBATCH --account=PZSXXXX
#SLURM already starts job in working directory
cd $SLURM_SUBMIT_DIR

#Set up software environment
module load intel
#Move input files to compute node
cp hello.c $TMPDIR

mpicc -O2 hello.c -o hello
srun ./hello > hello_results
#Copy results back to working directory
cp hello_results $SLURM_SUBMIT_DIR
```

Job setup information for SLURM

Commands to be run

Put all this into a text file!

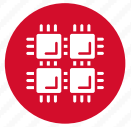


Submit & Manage Batch Jobs

Explanations	Torque	Slurm
Line that specifies the shell	No need	`#!/bin/bash`

Resource specification	`#PBS -l walltime=1:00:00`
	`#PBS -l nodes=2:ppn=40`
	`#PBS -N hello`
	`#PBS -A PZS0712`
		`#SBATCH --time=1:00:00`
		`#SBATCH --nodes=2 --ntasks-per-node=40`
		`#SBATCH --job-name=hello`
		`#SBATCH --account=PZS0712`

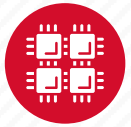
Variables, paths, and modules	`cd $PBS_O_WORKDIR` `module load intel`	`cd $SLURM_SUBMIT_DIR` `module load intel`
Launch and run application	`mpicc -O2 hello.c -o hello` `mpiexec ./hello > hello_results`	`mpicc -O2 hello.c -o hello` `srun ./hello > hello_results`



Job Submission Options

TORQUE/Moab directive	SLURM directive
`#PBS -N myjob`	`#SBATCH --job-name=myjob`
`#PBS -l walltime=1:00:00`	`#SBATCH --time=1:00:00`
`#PBS -l nodes=N:ppn=M`	`#SBATCH --nodes=N --ntasks-per-node=M`
`#PBS -l nodes=N:ppn=M:gpus=G`	`#SBATCH --nodes=N --ntasks-per-node=M` `#SBATCH --gpus-per-node=G`
`#PBS -l mem=Xgb`	`#SBATCH --memory=Xgb`
`#PBS -l software=pkg1+1%pkg2+4`	`#SBATCH --licenses=pkg1:1,pkg2:4`
`#PBS -l advres=rsvid`	`#SBATCH --reservation=rsvid`
`#PBS -o outfile`	`#SBATCH --output=outfile`

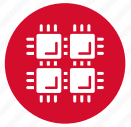
https://www.osc.edu/supercomputing/knowledge-base/slurm_migration



Job Environment Variables

TORQUE/Moab environment variable	SLURM environment variable
$PBS_JOBID	$SLURM_JOB_ID
$PBS_JOBNAME	$SLURM_JOB_NAME
$PBS_QUEUE	$SLURM_JOB_PARTITION
$PBS_O_WORKDIR	$SLURM_SUBMIT_DIR
$PBS_NODEFILE	$SLURM_JOB_NODELIST
$PBS_NUM_NODES	$SLURM_JOB_NUM_NODES
$PBS_NP	$SLURM_NTASKS
$PBS_NUM_PPN	$SLURM_TASKS_PER_NODE
$PBS_WALLTIME	$SLURM_TIME_LIMIT

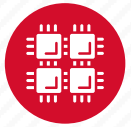
- Exhaustive list:
  - https://www.osc.edu/supercomputing/knowledge-base/slurm_migration/how_to_prepare_slurm_job_scripts



Submit & Manage Batch Jobs

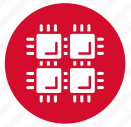
TORQUE/Moab	Slurm
qsub <jobscript>	sbatch <jobscript>
qdel <jobid>	scancel <jobid>
qhold <jobid>	scontrol hold <jobid>
qrls <jobid>	scontrol release <jobid>
qstat -u <user>	squeue -u <user>

Submit job
SLURM Response:
Submitted batch job 35484



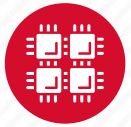
Scheduling Policies and Limits

- 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



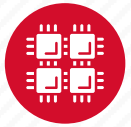
Waiting for Your Job To Run

- 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



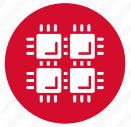
Interactive Batch Jobs

- 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`
  - `salloc -t 00:05:00 --ntasks-per-node=3`



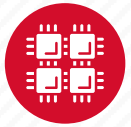
Batch Queues

- 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 on Ruby and Owens: -q debug



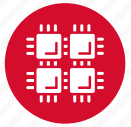
Parallel Computing

- 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 `MPI` for multiple nodes
- **Can't just request more nodes or cores and expect your job to run faster**



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/

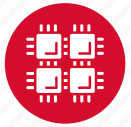
Office Hours: Virtual, every other Tuesday, 1:00pm – 4:00pm

OSC Events Calendar: https://www.osc.edu/events SLURM Migration 9/23

Ask.ci: https://ask.cyberinfrastructure.org/c/ohio-supercomputing/54

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