Ohio Supercomputer Center
An OH·TECH Consortium Member
An introduction to OSC services, hardware, and environment

May 20, 2021
“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

CY2020

- 34 Ohio Universities
- 59 Companies
- 33 Non-Profits and Government Agencies
- 124 Other Educational Institutions
- 5,256 Clients
- 209 Ohio Higher-Ed Courses
- 17 Training Opportunities
- 374 Trainees
Ohio Higher-Ed using OSC

CY2020
Ohio Higher-Ed Courses using OSC
CY2020
4,732 students, 209 courses, 78 departments, 21 universities

<table>
<thead>
<tr>
<th>Institution</th>
<th>Students</th>
<th>Courses</th>
<th>Dept.’s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Air Force Inst. of Tech.</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Bluffton University</td>
<td>10</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Bowling Green State U.</td>
<td>35</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Cedarville University</td>
<td>14</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Cleveland State U.</td>
<td>27</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Denison University</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Kent State University</td>
<td>48</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>Kenyon College</td>
<td>41</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Miami University</td>
<td>56</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Ohio State University</td>
<td>3,538</td>
<td>129</td>
<td>33</td>
</tr>
<tr>
<td>Ohio University</td>
<td>31</td>
<td>6</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Institution</th>
<th>Students</th>
<th>Courses</th>
<th>Dept.’s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stark State College</td>
<td>26</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>University of Akron</td>
<td>9</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>University of Cincinnati</td>
<td>747</td>
<td>27</td>
<td>10</td>
</tr>
<tr>
<td>University of Dayton</td>
<td>18</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>University of Mount Union</td>
<td>11</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>University of Toledo</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Urbana University</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Wittenberg University</td>
<td>7</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Wright State University</td>
<td>59</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Xavier University</td>
<td>45</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>
HPC Concepts
## Why Use HPC?

<table>
<thead>
<tr>
<th>Reason</th>
<th>Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Your simulations or analyses take too long on your personal computer</td>
<td>More (faster) cores</td>
</tr>
<tr>
<td></td>
<td>Multithreading</td>
</tr>
<tr>
<td></td>
<td>Multi-node parallelization (openmpi &amp; Rmpi)</td>
</tr>
<tr>
<td></td>
<td>GPU acceleration (NVIDIA’s CUDA)</td>
</tr>
<tr>
<td></td>
<td>Distributed computing (Apache Spark with Python and R)</td>
</tr>
<tr>
<td>The size of your data is too large to be contained (storage) or accessed (memory) on your computer</td>
<td>Large memory nodes: 768GB; 1.5TB, 3TB</td>
</tr>
<tr>
<td></td>
<td>Distributed computing (Apache Spark with R)</td>
</tr>
<tr>
<td></td>
<td>Compute node disk space: 1TB, 4TB, &amp; 24TB</td>
</tr>
<tr>
<td></td>
<td>Project storage: TBs range</td>
</tr>
<tr>
<td>You need a particular software or package for your work</td>
<td></td>
</tr>
</tbody>
</table>
What is the difference between your laptop and a supercomputer?
HPC Terminology

**Compute Node**
- Equivalent to a high-end workstation, part of a cluster

**Compute Cluster**
- A group of computers (nodes) connected by a high-speed network, forming a supercomputer

**Core**
- A processor (CPU), multiple cores per processor chip

**Graphical Processing Unit (GPU)**
- A separate multi-core processor that can handle many small calculations
Memory

Holds data that is being calculated on, as well as computational instructions

Memory types
- Shared memory is local to one node and several processes.
- Distributed memory is on multiple nodes.

Each core has an associated amount of memory
- Standard nodes: ~4 GB/core
- Huge memory nodes: ~75 GB/core
## Storage

Storage: different types of “disk” for different needs

- **Local disk in the compute node**
- **Project storage**
- **Shared scratch**  
  - Short-term storage only
- **Longer-term or archive**
Structure of a Supercomputer

Remote User Access

- **Login Nodes**
  - **Terminal**: access via ssh program on your computer
  - **Web portal**: access via ondemand.csc.edu

**Data Storage**
- **Home**, **Project**, **Scratch**

**Compute Nodes**
- 634 standard nodes
  - 12 cores per node
  - 48 GB memory
  - 812 GB local disk space

**Peak Performance**: 88 TF CPU + 66 TF GPU = 154 TF
Hardware Overview
Owens Compute Nodes

- **Terminal**: Access via ssh program on your computer
- **Login**: Access via ondemand.osc.edu
- **Web Portal**: Access via ondemand.osc.edu

**Owens Cluster**
- **Peak Performance**: ~750 TF CPU

**Nodes**
- **Home**
- **Project**
- **Scratch**
- **Compute Nodes**

**648 Standard Nodes**
- 28 cores per node
- Intel Xeon E5-2680 V4
- 128 GB memory
Owens Data Analytics Nodes

Peak Performance: ~750 TF CPU

16 large memory nodes
48 cores per node
Intel Xeon E5-4830 V3
1,536 GB memory
12 TB local disk space
Owens GPU Nodes

Owens Cluster

- Peak Performance: 706 TF CPU + ~750 TF GPU
- 100 Gb/sec Infiniband Network (EDR)

- login nodes
- 100 GPU nodes
  - 160 NVIDIA Pascal P100 GPUs
  - 1 each on 160 standard nodes

- shared data storage
- compute nodes

terminal
access via
ssh program
(owens.esc.edu)

portal
access via
web browser
(ondemand.esc.edu)

login
Owens Cluster Specifications

Peak Performance: 706 TF CPU + ~750 TF GPU

- 648 standard nodes
  - Intel Xeon E5-2680 V4 (Broadwell)
  - 28 cores per node
  - 128 GB memory
  - 1.5 TB local disk space

- 6 debug nodes
  - standard nodes
  - 1 hour walltime limit

- 160 GPU nodes
  - 160 NVIDIA Pascal P100 GPUs
  - 1 each on 160 standard nodes

- 16 large memory nodes
  - Intel Xeon E5-4830 V3 (Haswell)
  - 48 cores per node
  - 1.5 TB memory
  - 24 TB local disk space

- 100 Gb/sec Infiniband Network (EDR)

Shared data storage: home, project, scratch

Terminal access via seh program (owens.osu.edu)
Portal access via web browser (ondemand.osu.edu)

https://www.osc.edu/resources/technical_support/supercomputers/owens
Pitzer Cluster Specifications

- **Terminal access via ssh program (pitzer.osc.edu)**
- **Portal access via web browser (ondemand.osc.edu)**

**224 standard nodes**
- 2x Intel Xeon Gold 6148 (Skylake)
- 40 cores per node
- 192 GB memory
- 1 TB local disk space

**32 GPU nodes**
- 2x NVIDIA Volta V100 GPUs
- 40 cores per node
- 384 GB memory
- 1 TB local disk space

**4 huge memory nodes**
- 4x Intel Xeon Gold 6148 (Skylake)
- 80 cores per node
- 3 TB memory
- 1 TB local disk space

**Peak Performance:** ~799 TF CPU + ~499 TF GPU

**Shared data storage**
- home
- project
- scratch

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
File Systems at OSC

Remote User Access

- Terminal: access via ssh program on your computer
- Web portal: access via ondemand.csc.edu

Login Nodes

- Peak Performance: 88 TF CPU + 66 TF GPU = 154 TF

Data Storage

- home
- project
- scratch

Compute Nodes

- 634 standard nodes
  - 12 cores per node
  - 48 GB memory
  - 812 GB local disk space
Research Data Storage

- **Home**
  - Store your files here, backed up daily
  - 500GB per user
- **Project/ESS**
  - Available to Project PIs by request; shared by all users on a project, backed up daily
  - 1-5TB standard request
- **Scratch**
  - Faster I/O than Home or Project
  - Temporary storage, not backed up
- **Compute Nodes**
  - Storage on compute nodes, for use during your batch job
  - All data purged when job quits
- **Archive**
  - Long term storage, by request

<table>
<thead>
<tr>
<th>Filesystem (Reference)</th>
<th>Quota</th>
<th>Network</th>
<th>Backed-Up?</th>
<th>Purged?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home ($HOME)</td>
<td>500GB</td>
<td>10 GB/s</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Project (/fs/project or /fs/ess)</td>
<td>By request (Typically 1-5 TB)</td>
<td>50 GB/s</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Scratch (/fs/scratch)</td>
<td>100TB</td>
<td>100 GB/s</td>
<td>No</td>
<td>Yes – 90 days</td>
</tr>
<tr>
<td>Compute ($TMPDIR)</td>
<td>Varies (~1 TB)</td>
<td>Varies</td>
<td>No</td>
<td>Yes – when job completes</td>
</tr>
</tbody>
</table>

https://www.osc.edu/supercomputing/storage-environment-at-osc/storage-hardware/overview_of_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
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
  - Software and Activities
  - Hardware and Operations
- Get involved!
  - Twice a year - April and October
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 & Monitoring
- 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 & Programming environment
Third party applications

- **Parallel programming software (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 Software Environment

<table>
<thead>
<tr>
<th>Action</th>
<th>Command/Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>What modules do you have loaded?</td>
<td>module list</td>
</tr>
<tr>
<td>What modules are available?</td>
<td>module spider or module avail</td>
</tr>
<tr>
<td>Multiple versions of the same software</td>
<td>module avail intel</td>
</tr>
<tr>
<td>Add a software module to your environment</td>
<td>module load cuda</td>
</tr>
<tr>
<td>Remove a software package from your environment</td>
<td>module unload intel</td>
</tr>
<tr>
<td>Load a different software version</td>
<td>module swap intel intel/13.1.3.192</td>
</tr>
</tbody>
</table>
Batch Processing
Why do supercomputers use queuing?

- **Terminal**: Access via ssh program on your computer
- **Web Portal**: Access via ondemand.osc.edu

**Cluster**

- **Job Queue**: Submit a batch script to the queue using the "qsub" command.
- **Scheduler**: A job scheduler manages the queue to ensure jobs run efficiently.

**Storage**

- **Home**
- **Project**
- **Scratch**

**Compute Nodes**

**Active Jobs**: Multiple jobs run using a variety of nodes depending on the time constraints, number, and type of nodes requested.
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 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

Put all this into a text file!
# Submit & Manage Batch Jobs

<table>
<thead>
<tr>
<th>SLURM Directive</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sbatch &lt;jobscript&gt;</td>
<td>Submit job script</td>
</tr>
<tr>
<td>scancel &lt;jobid&gt;</td>
<td>Cancel a job</td>
</tr>
<tr>
<td>scontrol hold &lt;jobid&gt;</td>
<td>Put job on hold</td>
</tr>
<tr>
<td>scontrol release &lt;jobid&gt;</td>
<td>Release job from hold</td>
</tr>
<tr>
<td>squeue -u &lt;user&gt;</td>
<td>View information about job(s) of a user</td>
</tr>
</tbody>
</table>

Submitted 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)
  - `sinteractive -N 1 -n 4 -t 00:10:00 -J test`
Batch Queues

• The two 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)
## Parallel Computing

Each processor is fast, but real speed comes from using multiple processors

<table>
<thead>
<tr>
<th>Multithreading</th>
<th>Use multiple cores on a single node</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Shared memory</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Message passing interface (MPI)</th>
<th>Use one or multiple nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Distributed memory</td>
</tr>
</tbody>
</table>
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
- Use `mpiexec` 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

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

System updates

• Read Message of the Day on login
• Follow @HPCNotices on Twitter
Questions?
# Ohio Academic Price Sheet

<table>
<thead>
<tr>
<th><strong>High Performance Computing Services</strong></th>
<th><strong>Standard Nodes</strong></th>
<th><strong>Big Memory Nodes</strong></th>
<th><strong>Add-on GPU</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>per node hr</td>
<td>per core hr</td>
<td>per node hr</td>
</tr>
<tr>
<td>Owens Cluster</td>
<td>$0.08</td>
<td>$0.003*</td>
<td>$0.19</td>
</tr>
<tr>
<td>Pitzer Cluster</td>
<td>$0.12</td>
<td>$0.003*</td>
<td>$0.32</td>
</tr>
<tr>
<td>Pitzer Expansion</td>
<td>$0.14</td>
<td>$0.003*</td>
<td>$0.19</td>
</tr>
</tbody>
</table>

*Current subsidies allow for reduced costs. Original per core hour cost is $0.014. Costs subject to change.*

Monthly billing is based on usage of nodes/cores to the nearest minute.

## Data Storage & Transfer Services

<table>
<thead>
<tr>
<th></th>
<th>Price per TB per month</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home directories, parallel scratch and network transfer</td>
<td>$0</td>
</tr>
<tr>
<td>Project storage (high performance, high availability file system, includes backup)</td>
<td>$1.60</td>
</tr>
</tbody>
</table>

Monthly billing is based on the allocated storage quota to the nearest half TB.