

NAIRR Pilot

National Artificial Intelligence
Research Resource Pilot



NAIRR Pilot

National Artificial Intelligence
Research Resource Pilot

How to use an HPC system
Mary Thomas
April 2, 2024, Track 1 - Beginner
AI Workshop Denver, CO April 2-3, 2025





Session: How to Use an HPC System (90 minutes)

This session introduces participants to the fundamentals of using High-Performance Computing (HPC) systems for research and AI applications. Key topics include:

1. Understanding HPC Architecture:

- Overview of research computing resource structures, including nodes, processors, and storage.

2. Discovering Available Software and Tools:

- Using tools like the **ACCESS Recommender** and **ACCESS Software Recommender** to identify suitable software for your projects.

3. Submitting Jobs to an HPC System:

- Step-by-step guidance on using a job scheduler for job submission, along with demonstrations of multiple methods.
- Reusable example scripts for efficient job setup and execution.

4. Interactive HPC Access:

- Using **Open OnDemand** and platforms like **JupyterHub** for streamlined, user-friendly interaction with HPC resources.

Participants will gain practical skills for navigating HPC systems, leveraging software tools, and optimizing workflows for research and AI projects.



1. Understanding HPC Architecture

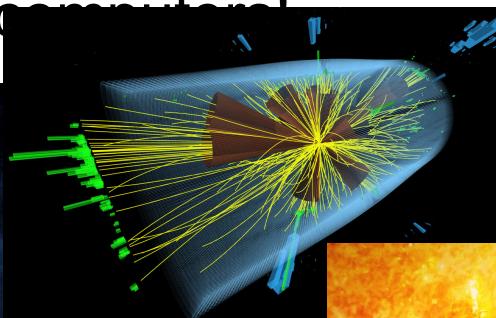
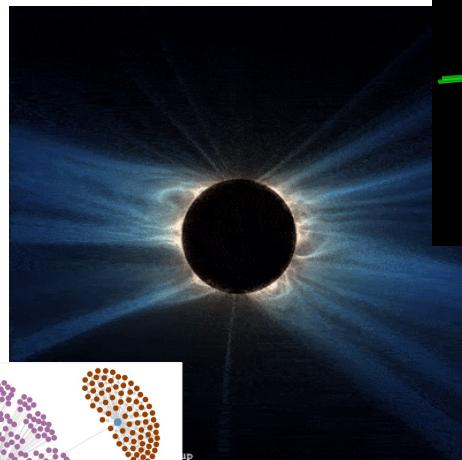
NAIRR Pilot

National Artificial Intelligence
Research Resource Pilot



Pelagic fish communities Shapes
(Jerome Guiet, UCLA)

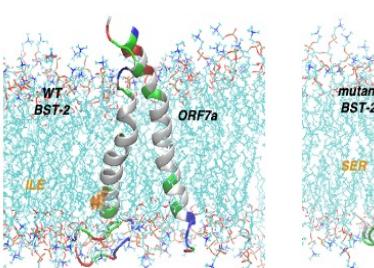
Scaling Jobs on Supercomputers



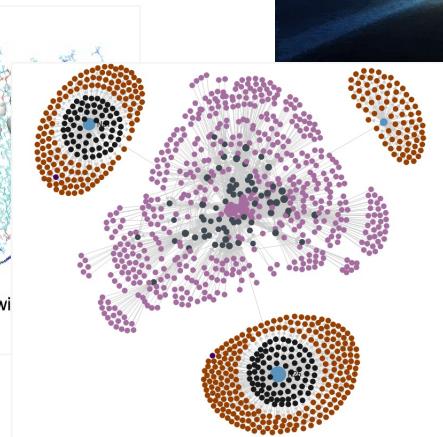
Compact Muon Solenoid
(CMS) experiment at the LHC,
CERN. Image courtesy of
CMS Collaboration; Mc
Cauley, Thomas



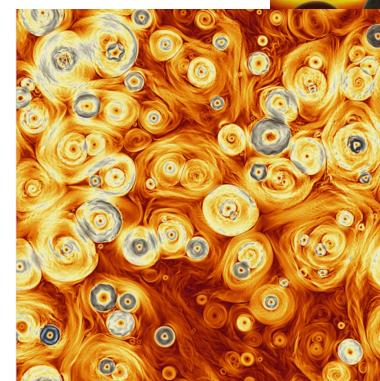
Model of Both Inner and Outer Solar System
M. S.Clement (Carnegie Institution for Science)



Jeff Klauda /U. Maryland
Interactions between SARS-CoV-2 ORF7a with wild-type (left) and mutant (right) BST-2



Sample of Internet
structure from
CAIDA data (Mark
Burgess, 12/16/21)



Modeling the sun's corona,
Alfred Mallet (UC Berkeley)

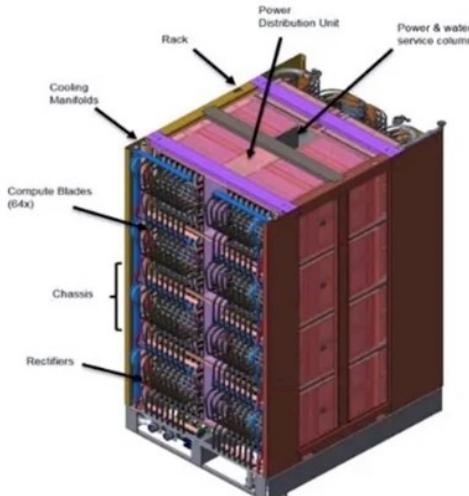


HPC System: ORNL FRONTIER



System

- 2 EF Peak DP FLOPS
- 74 compute racks
- 29 MW Power Consumption
- 9,408 nodes
- 9.2 PB memory
(4.6 PB HBM, 4.6 PB DDR4)
- Cray Slingshot network with dragonfly topology
- 37 PB Node Local Storage
- 716 PB Center-wide storage
- 4000 ft² foot print



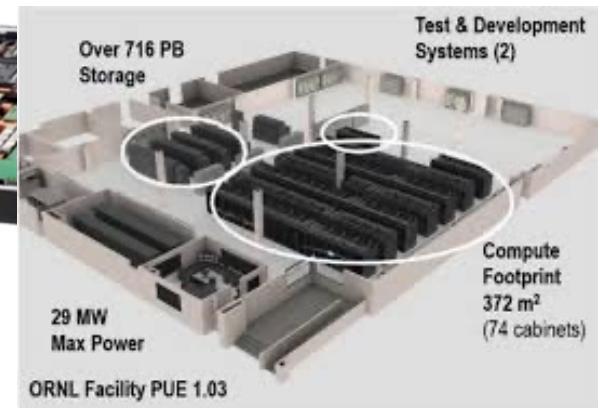
All water cooled, even DIMMs and NICs

AMD node

- 1 AMD “Trento” CPU
- 4 AMD MI250X GPUs
- 512 GiB DDR4 memory on CPU
- 512 GiB HBM2e total per node
(128 GiB HBM per GPU)
- Coherent memory across the node
- 4 TB NVM
- GPUs & CPU fully connected with AMD Infinity Fabric
- 4 Cassini NICs, 100 GB/s network BW

Compute blade

- 2 AMD nodes

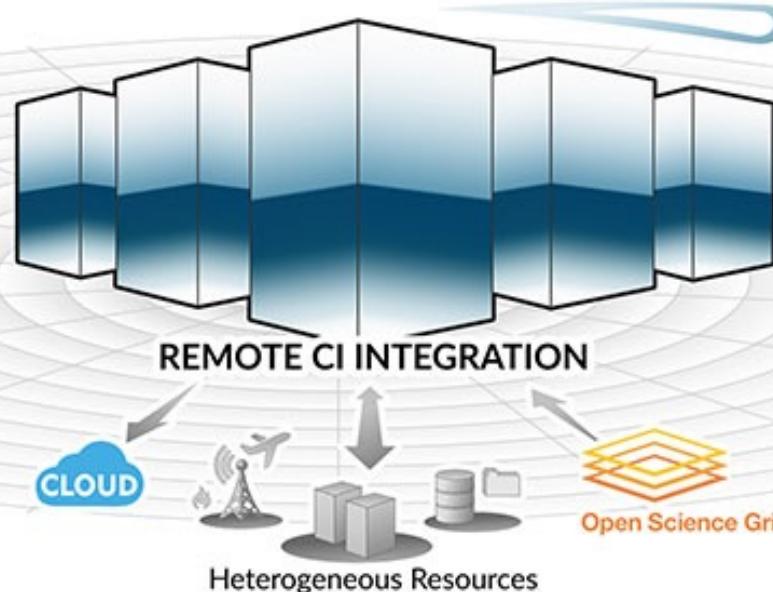




HPC Supercomputer: Expanse

HPC RESOURCE

13 Scalable Compute Units
728 Standard Compute Nodes
52 GPU Nodes: 208 GPUs
4 Large Memory Nodes



LONG-TAIL SCIENCE

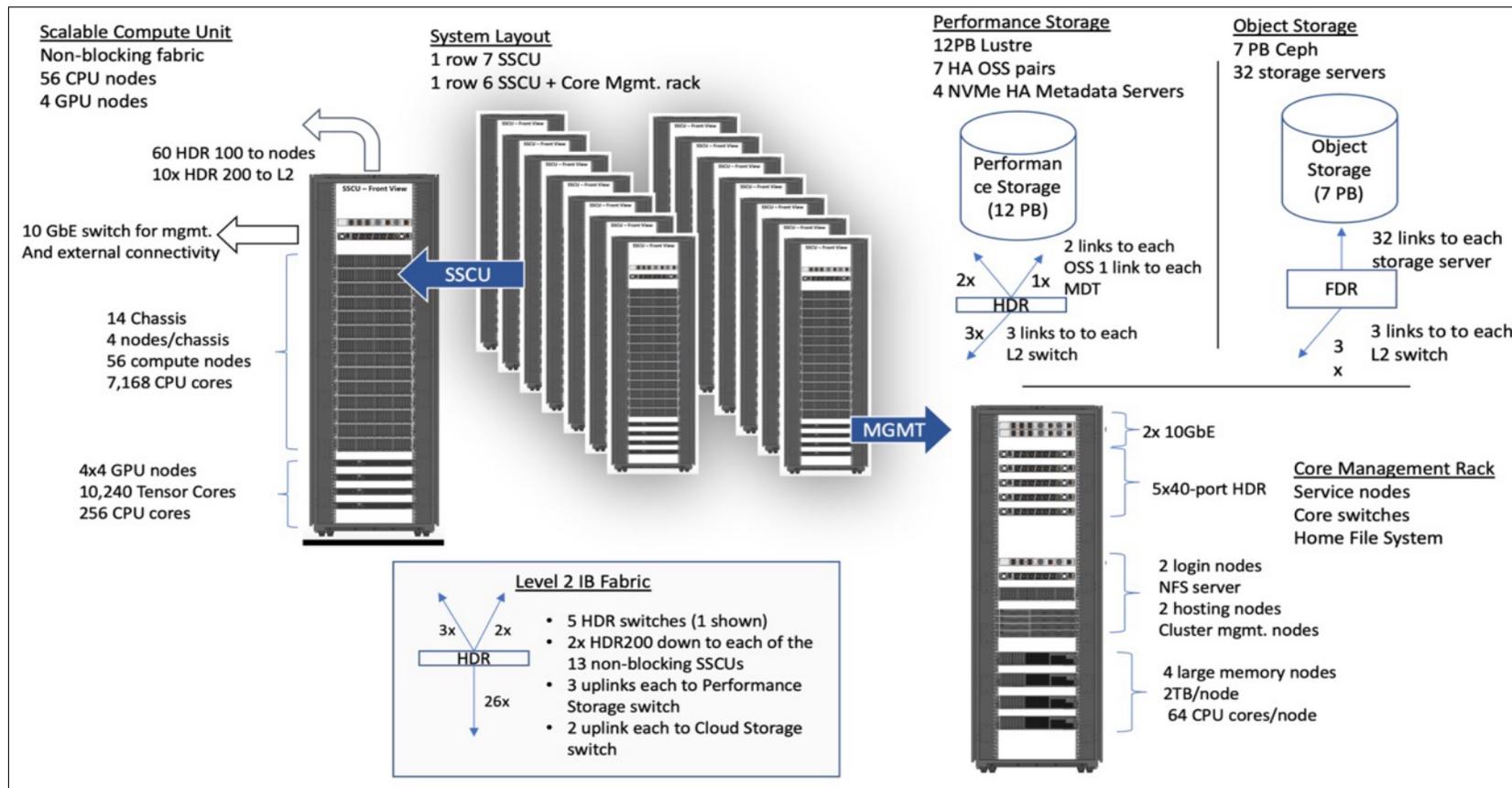
Multi-Messenger Astronomy
Genomics
Earth Science
Social Science

INNOVATIVE OPERATIONS

DATA CENTRIC ARCHITECTURE
12PB Perf. Storage: 140GB/s, 200k IOPS
Fast I/O Node-Local NVMe Storage
7PB Ceph Object Storage
High-Performance R&E Networking

Composable Systems
High-Throughput Computing
Science Gateways
Interactive Computing
Containerized Computing
Cloud Bursting

HPC System Architecture: Expanse @ SDSC





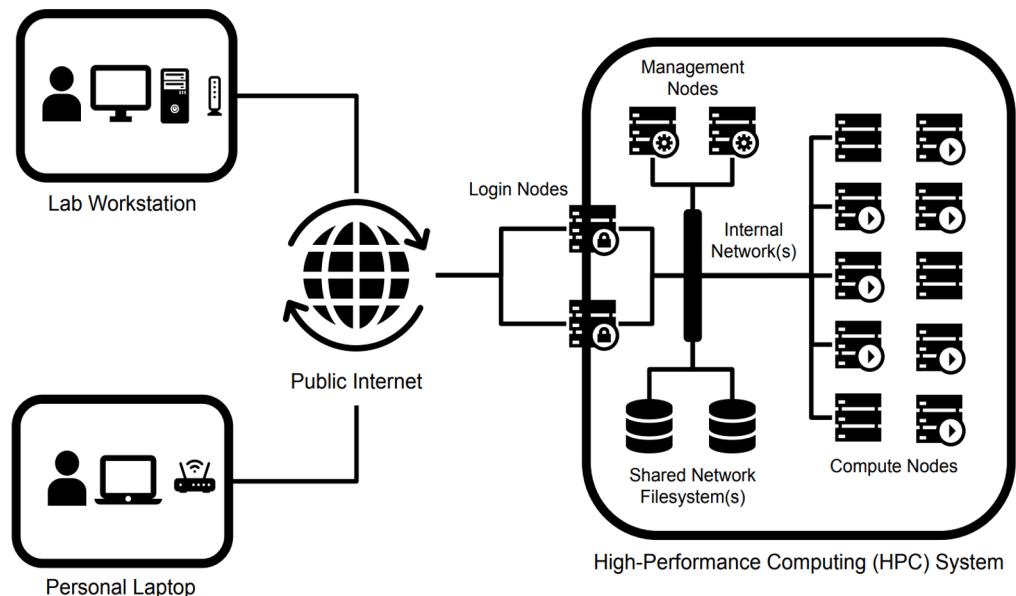
HPC System Architecture: Conceptual Model

Login node(s): Provide remote access to an HPC system; use only for simple tasks such as editing files, limited data transfers to and from the system, and batch job submission

Compute nodes: Run computational workloads: simulations, data analysis and visualization

Internal Network(s): Provide high-bandwidth, low-latency communication between compute nodes ; access to shared (parallel) filesystems; system management

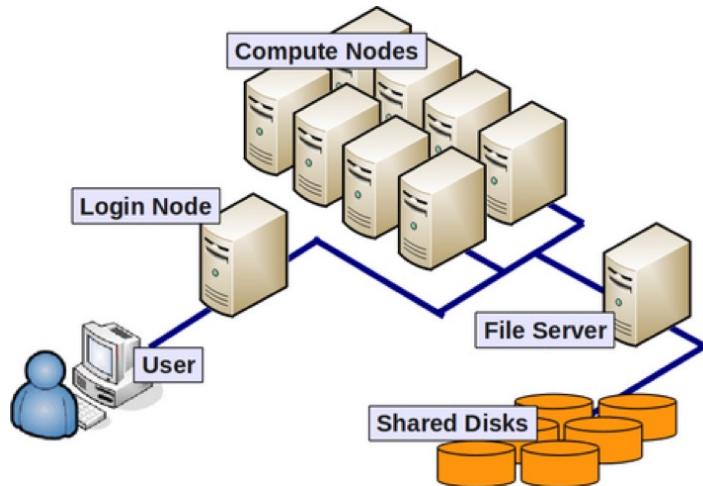
Shared Network Filesystem(s): Provide input/output (I/O) access to data storage systems from any compute node



Management node(s): Run core system services such as cluster management software, system monitoring software, *batch job scheduler*, etc



Login and Compute Nodes



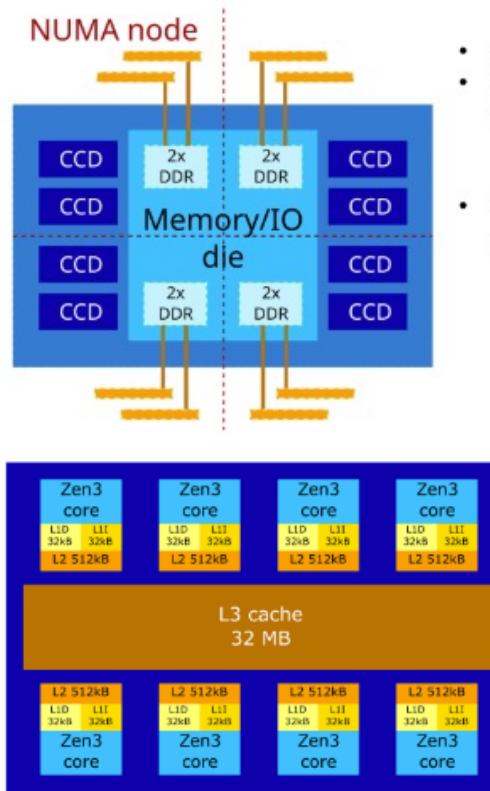
https://www.advancedclustering.com/act_systems/hpc-compute-node/

https://hbctraining.github.io/Intro-to-shell-fasrc-flipped/lessons/08_HPC_intro_and_terms.html



Expanse CPU Nodes

- 13+ SDSC Scalable Compute Units (SSCU, racks)
 - 56 CPU nodes
 - 4 GPU nodes (V100+H100)
- 728+ AMD EPYC 7742 AMD Rome Standard Compute Nodes
- 128 cores/node
- 256 GB DRAM/node
- NVME 1TB/node

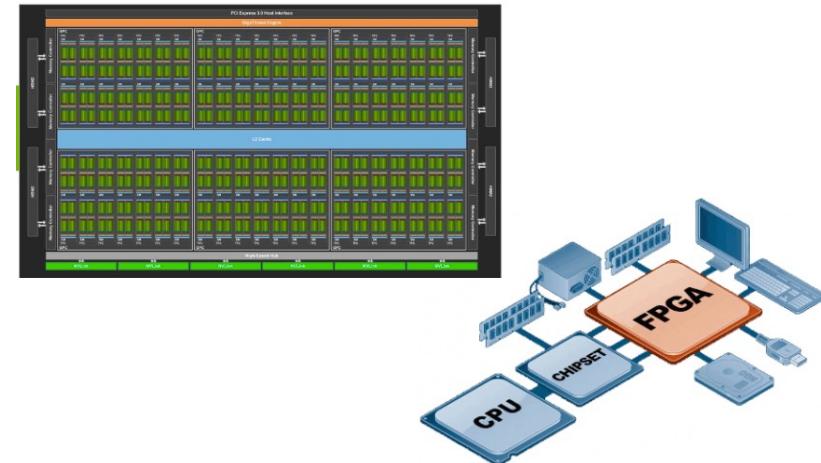


- 8 CCDs or 8 L3 cache regions
- Memory/IO die logically split into 4 NUMA domains with
 - 2 CCDs (16 cores)
 - 2 DDR4 controllers
- Memory/IO die also provides the PCIe links and intersocket links
- Building block: a Core Complex Die (CCD)
 - 8 cores
 - Each core has private L1 and L2 caches
 - L3 cache shared
- Instruction set equivalent to Intel Broadwell generation
 - AVX2+FMA, no AVX-512

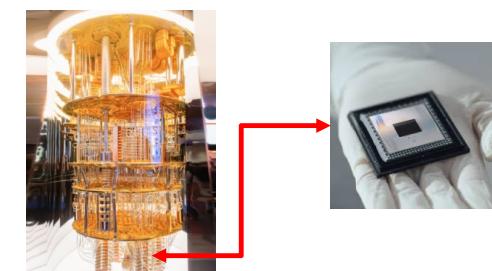


Compute Accelerators for AI: GPU, FPGAs, QC

- Different accelerators work best for different application spaces
- GPU: Expanse: 52 GPU Nodes; 4 GPUs/node;
 - 384 GB CPU DRAM/node;
 - 80 SMs (Streaming Multiprocessor)
 - NVIDIA Tesla V100 SMX2: 32GB memory/GPU;
 - 21 B transistors; 5120 CUDA cores
 - 640 Tensor cores
- Field Programmable Gate Arrays (FPGA):
 - Three types of modules: I/O blocks; Switch Matrix/Interconnection Wires; Configurable logic blocks (CLB);
 - Features: customized for application; real-time data processing; parallel processing
 - Emerging AI popularity
- Quantum Computers: in the future
 - <https://quantumai.google/>



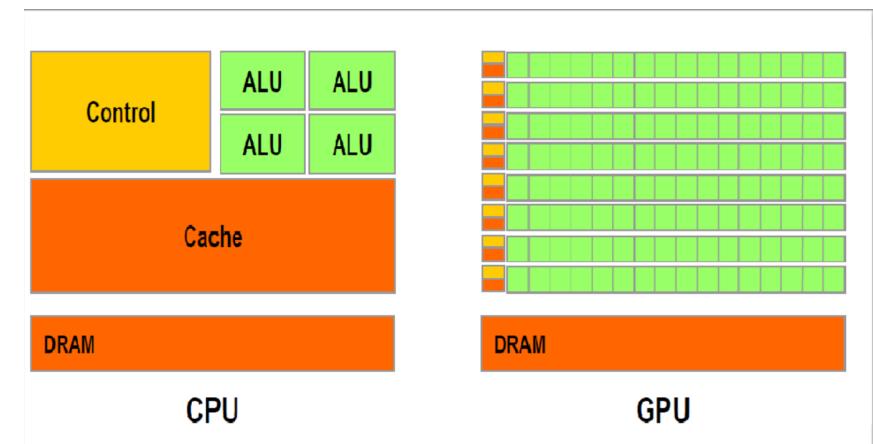
<https://www.elprocus.com/fpga-architecture-and-applications/>





Difference between CPUs and GPUs and why AI likes GPUs

- CPU (compute processor unit) is
 - designed to handle a wide-range of tasks quickly (as measured by CPU clock speed),
 - but are limited in the concurrency of tasks that can be running.
 - Parallelism achieved with multiple cores and/or nodes; data exchanged
- GPU (Graphic Processing Unit) is
 - designed to quickly render high-resolution images and video concurrently;
 - can perform parallel operations on multiple sets of data; but need to get data onto the GPU.
 - Designed for highly parallel computations and were also referred to as throughput processors.
 - GPU-accelerated applications offload these time-consuming routines and functions (also called hotspots)
- GPUs dedicate most of their transistors for data processing while CPUs also need to reserve die area for big caches, control units, and so on. CPU processors work on the principle of minimizing latency within each thread while GPUs hide the instruction and memory latencies with computation. Figure 3 shows the difference in computation threads.



<https://www.heavy.ai/technical-glossary/cpu-vs-gpu>

<https://developer.nvidia.com/blog/cuda-refresher-reviewing-the-origins-of-gpu-computing/>



Expanse File Systems

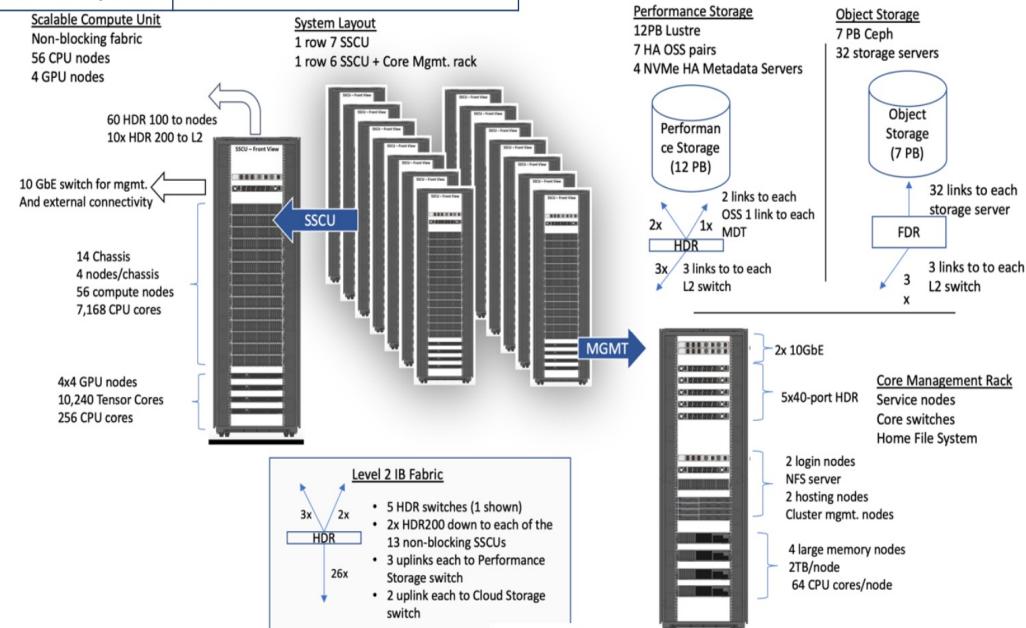
- Home directories (/home/\$USER) – 8 week rolling backup.
 - Login nodes: 100GB limit;
 - Use for source trees, binaries, and small input files.
 - Not good for large scale I/O.
- Lustre filesystems: Good for scalable large block I/O
 - /expanse/lustre/scratch/\$USER/temp_project
 - 2.5PB; peak performance: 100GB/s. Good for storing large scale scratch data during a job.
 - /expanse/lustre/projects/-
 - 2.5PB, peak performance: 100 GB/s. Long term storage.
 - Not good for large # of small files or small block I/O
- Local node “scratch” (SSD) filesystems
 - /scratch local to each native compute node – 210GB on regular compute nodes, 285GB on GPU, large memory nodes, 1.4TB on selected compute nodes.
 - SSD location is good for writing small files and temporary scratch files. Purged at the end of a job.
- Webinar on Data Management and File Systems:
 - https://www.sdsc.edu/event_items/202110_ExpanseWebinar-M.Shantharam.html



Expanse Storage and File Systems

System Component	Configuration
<i>AMD EPYC (Rome) 7742 Compute Nodes</i>	
Node count	728
Clock speed	2.25 GHz
Cores/node	128
Total # cores	93,184
DRAM/node	256 GB
NVMe/node	1 TB
<i>NVIDIA V100 GPU Nodes</i>	
Node count	52
Total # GPUs	208
GPUs/node	4
GPU Type	V100 SMX2
Memory/GPU	32 GB
CPU cores; DRAM; clock (per node)	40; 384 GB; 2.5 GHz;
CPU	6248 Xeon
NVMe/node	1.6TB
<i>Large Memory Nodes</i>	
Number of nodes	4
Memory per node	2 TB
CPUs	2x AMD 7742/node;

Storage	
Lustre file system	12 PB (split between scratch & allocable projects)
Ceph file system	7 PB Coming soon
Home File system	1 PB





Expanse Storage: Global Lustre Filesystem

- Global parallel Lustre filesystems
 - 12 PB Lustre parallel file system
 - 7 PB Ceph Object Store system
 - 140 GB/second performance storage.
- Accessible from all compute and GPU nodes.
 - Lustre Expanse scratch filesystem: limited to 2 millions files/user
 - /expanse/lustre/scratch/\$USER/temp_project
 - Lustre NSF projects filesystem: /expanse/lustre/projects/
- Best for scalable large block I/O
- Not good for large # of small files or small block I/O. Please. Don't. Do. This.
 - Contact help@xsede.org for guidance on how to handle this



Expanse Storage: local node/scratch

- /scratch is local to each native compute node (SSD)
- Latency to SSDs is several orders of magnitude lower than that for spinning disk (<100 microseconds vs. milliseconds)
- Path changes with Job: /scratch/\$USER/job_\${SLURM_JOB_ID}
- SSD location is good for writing small files and temporary scratch files. Purged at the end of a job.

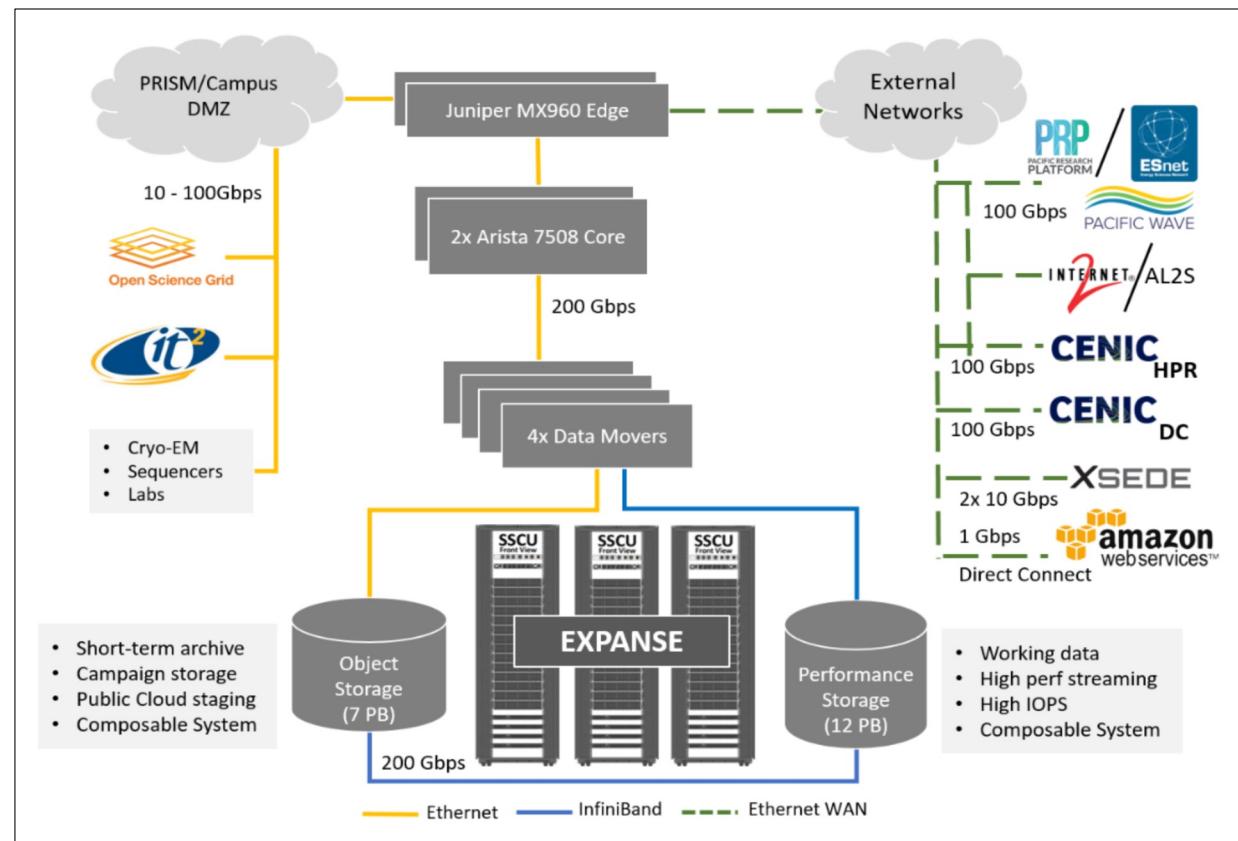
```
[username@exp-7-59 MPI]$ ll /scratch/ username /
total 4
drwx----- 2 username root 4096 Sep 16 01:55 job_5826715
[username @exp-7-59 MPI]$ ll /scratch/ username /job_5826715/
total 0
[username @exp-7-59 MPI]$ !sq
squeue -u username
      JOBID PARTITION    NAME   USER ST       TIME  NODES NODELIST(REASON)
  5826715 gpu-debug    bash  username  R     0:15    1 exp-7-59
[username@exp-7-59 MPI]$
```

Partition	Space
compute,shared	1 TB
gpu, gpu-shared	1.6TB
large-shared	3.2 TB



Expanse Network

- Infiniband:
 - 200 Gbps internal
 - 100 Gbps external
- 256 GB/node (DRAM)
- Multiple storage devices & types
 - Local node (1TB)
 - Scratch file system
 - Luster (PB)
 - Ceph Object
- High-speed interconnection networks
 - Local, remote
- Running in parallel
- Optimized software





Expanse access via several mechanisms

- **Command line** using an ACCESS-wide password or ssh-keys with TOTP:
login.expanse.sdsc.edu
- **Web-based** access via the [Expanse User Portal](https://portal.expanse.sdsc.edu):
portal.expanse.sdsc.edu
- **Customized services**





Hands On: Logging onto *Expanse*: `login.expanse.sdsc.edu`

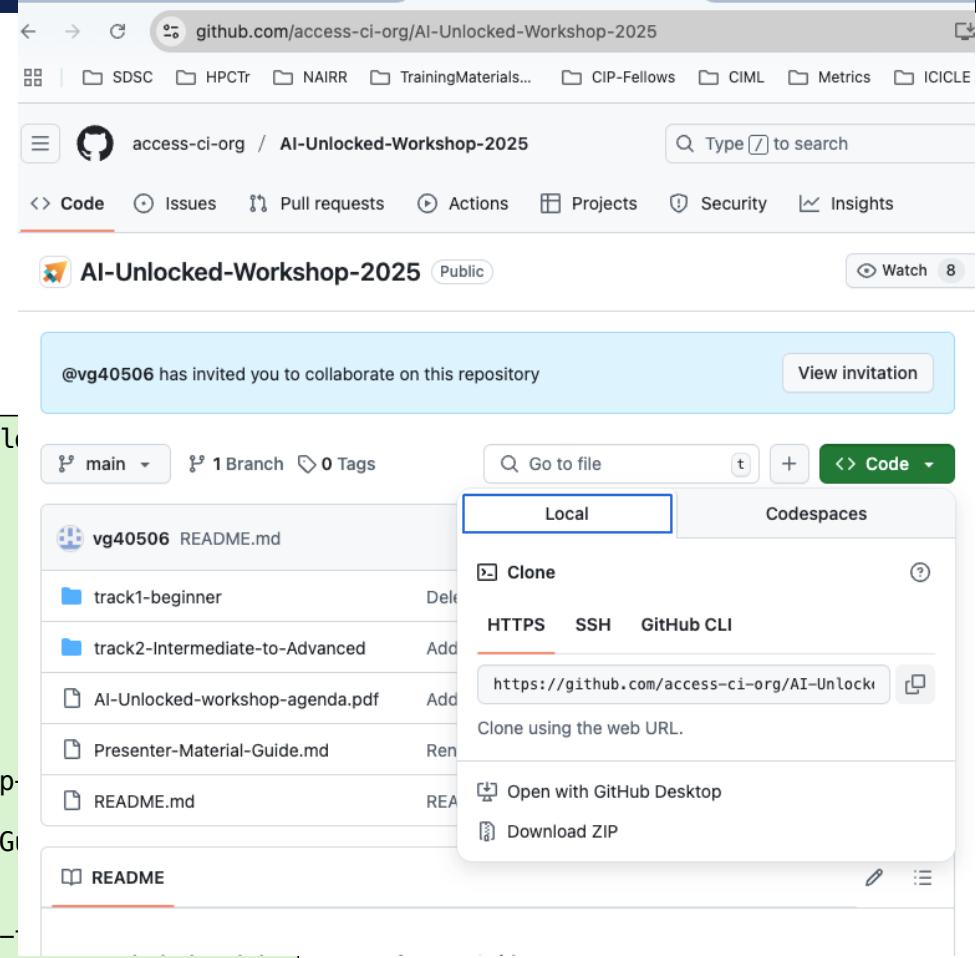
- *Expanse* supports access via:
 - Command line using an **ACCESS-wide password or ssh-keys with TOTP**
 - Web-based access via the **Expanse User Portal**. While CPU and GPU resources are allocated separately, the login nodes are the same.
- Log onto *Expanse*:
 - `ssh <your_username>@login.expanse.sdsc.edu`
 - `ssh -l <your_username> login.expanse.sdsc.edu`
 - To set up 2FA authorization with TOTP, see the *Expanse* instructions here:
 - https://www.sdsc.edu/systems/expanse/user_guide.html#narrow-wysiwyg-2
- Make sure you can clone the workshop repo:
 - <https://github.com/access-ci-org/AI-Unlocked-Workshop-2025>
- Browse around the *Expanse* system: what applications and examples are in the /cm directories?
- For basic HPC Linux skills, see: https://hpc-training.sdsc.edu/basic_skills/



Hands-on: Cloning the Repo

- Access the GitHub Repo
 - Identify the green Code box & copy the HTTPS clone link
- Logon onto Expanse
- Run git clone command to install AI Unlocked repo

```
[mthomas@login02]$ git clone https://github.com/access-ci-org/AI-Unlocked-Workshop-2025.git
Cloning into 'AI-Unlocked-Workshop-2025'...
remote: Enumerating objects: 309, done.
remote: Counting objects: 100% (147/147), done.
[SNIP]
Resolving deltas: 100% (144/144), done.
Updating files: 100% (40/40), done.
[mthomas@login02]$ ls -al AI-Unlocked-Workshop-2025/
drwxr-xr-x 5 mthomas use300 8 Mar 31 22:47 .
drwxr-xr-x 9 mthomas use300 9 Mar 31 22:47 ..
-rw-r--r-- 1 mthomas use300 113513 Mar 31 22:47 AI-Unlocked-workshop-
drwxr-xr-x 8 mthomas use300 13 Mar 31 22:47 .git
-rw-r--r-- 1 mthomas use300 6462 Mar 31 22:47 Presenter-Material-G-
-rw-r--r-- 1 mthomas use300 3299 Mar 31 22:47 README.md
drwxr-xr-x 9 mthomas use300 10 Mar 31 22:47 track1-beginner
drwxr-xr-x 8 mthomas use300 10 Mar 31 22:47 track2-Intermediate-
```





2. Discovering Available Software and Tools



Learn About NAIRR HPC/CI Systems, Software, Tools

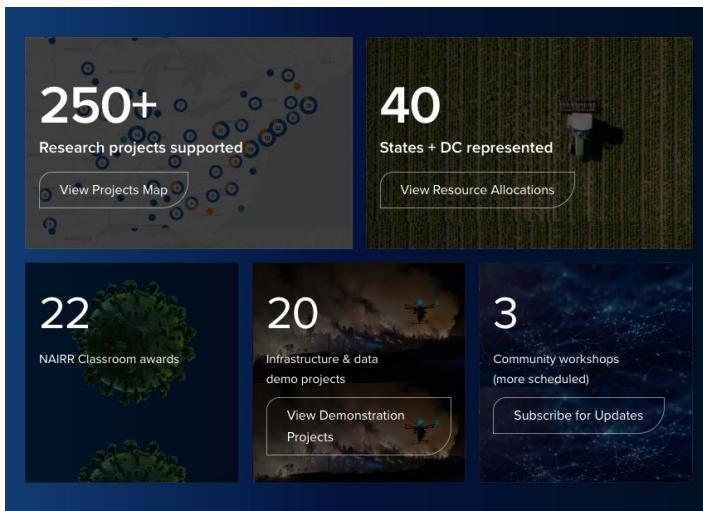
- National:
 - NAIRR Pilot Website: <https://nairrpilot.org/>
 - NSF ACCESS Portal
- Local HPC Center:
 - HPC System Information
 - User Guides
- On Host:
 - Logon and look around the system
 - Run local utilities to get system info
 - Learn about software with tools like modules, etc.



National Artificial Intelligence
Research Resource Pilot



NAIRR Pilot Website: <https://nairrpilot.org/>



NAIRR Pilot National Artificial Intelligence
Research Resource Pilot

Open Data, Models, and More

This list does not include allocatable resources for research or education/teaching; please see the [Researcher Call](#) and [Classroom/Educator Call](#) for those resources.

Current Opportunities

NAIRR Pilot Resource Requests to Advance AI Research

CyberInfrastructure Help / NAIRR Pilot Ticketing System

NAIRR NAIRR Pilot Ticketing System

Welcome! You can raise a request for NAIRR Pilot support using the options provided.

What can we help you with?



NAIRR Researcher Support Request

Researcher support requests related to allocations, resources, or training/classrooms.

Submit a ticket

Please include a short summary and a details description below

Required fields are marked with an asterisk *

Email confirmation to *

Department of Energy (DOE), and
for the research community
resources for projects related to
24, until the end of the NAIRR
cts will be awarded for twelve

ACCESS Portal: rich suite of services

Get Started

ACCESS is a large, distributed ecosystem. We want to make it easy for you to get started. We've compiled information and quick links just for you.

- I'm a researcher**
Get cutting-edge cyberinfrastructure for your research.
- I'm an educator**
Bring supercomputing into your classroom.
- I'm a graduate student**
Learn how to become eligible for ACCESS allocations.
- I'm a resource provider**
Manage and optimize your resource.
- I represent a program or organization**
See what ACCESS can do for your research community.

ALLOCATIONS MENU

My Projects

ACCESS Q&A Bot

Hello! What can I help you with?

Discover: Feb 26, 2025 to Feb 25, 2026

Overview Credits + Resources Users + Roles History

Resource	Status	Balance	End Date	My Username
Delta CPU	Active	15K of 15K Core-hours remaining (100%)	Feb 25, 2026	thomasm

Questions should stand alone and not refer to previous ones.

Resources

ACCESS provides advanced computing resources **at no cost** to researchers and educators.

- Browse resources**
Filter resources to find the best match for your research.
- Ask a question**
You have resource questions. Our Q&A bot has answers!
Requires ACCESS Login
- Get suggestions**
Fill out a form to get
- Read news**
See the latest news

Compute & Storage Resources

ACES

Texas A&M University GPU
CPU Compute ACCESS On-Demand Science Gateway support

Acceleration Computing for Everyone

REQUEST NEW PROJECT • GET HELP •

MATCH Services

ACCESS MATCH Services connects researchers with experts to help you select the right system, run on a supercomputer, and solve basic code and research problems.

MATCH Plus

- Help from an experienced expert
- Short engagements (typically 1-5 sessions with a mentor)
- Free

We will match you with an expert based on your needs.

Solve research code, transition HPC or introductory

REQUEST AN ENGAGEMENT

Steps in a MATCH Plus Engagement

- An ACCESS community member identifies a need for support and submits an engagement request.
- MATCH staff will work with you to scope the engagement and match you with an appropriate expert.

SUPPORT / TOOLS / XDMOD

XDMoD

Analyze and improve your allocation usage

LAUNCH XDMOD

Total CPU Hours by User Institution (Top 10)

Total CPU Hours by NSF Directorate

ACCESS XDMoD

About

Using XDMoD

Events & Trainings

Upcoming Events Past Events My Events

Wed 03/19/25 - 11:00 AM - 12:00 PM PDT
19 Mar Neocortex Office Hours
zoom machine-learning neural-networks neocortex

Thu 03/20/25 - 09:00 AM - 10:00 AM PDT
20 Mar CCMNet Monthly All Hands Meeting
Zoom community-outreach mentorship professional-development workforce-deve

Thu 03/20/25 - 11:00 AM - 12:00 PM PDT
20 Mar Anvil Support Hour
Zoom NAIRR-pilot anvil community-outreach

Thu 03/20/25 - 11:00 AM - 12:30 PM PDT
20 Mar COMPLEXCS: Interactive Computing
The event will be held remotely.
cloud-computing cloud-storage computer-graphics visualization hpc computer-science data-science open-demand hpc-getting-started interactive-mode



ACCESS Software Documentation Service

Software Documentation Service

Disclaimer: This tool is a work in progress. Fields marked with a sparkles emoji (✨) have largely been generated by AI. Additional information will continue to be added in future updates. Users are encouraged to respond via the 'Report Issue' and 'Provide Feedback' buttons to provide missing information, report errors, or suggest corrections.

25 ▾
entries per pageShow/Hide Columns ▾Show Search ToolsHover your mouse to the edge of the table to scrollSearch:

[Report Issue](#)[Provide Feedback](#)

Software	RP Name	Software Description	✨ AI Description
7z	Ookami	7-Zip is a file archiver with a high compression ratio. Description Source: https://www.7-zip.org/	7-Zip is a file archiver with a high compression ratio. It supports several archive formats and can be used to compress and decompress files efficiently.
abacas	Anvil	Abacas is a tool for algorithm based automatic contiguation of assembled sequences.	ABACAS is a tool for rapid bacterial genome contig assembly. It takes contigs produced by an assembler and scaffolds them to generate a pseudo-chromosome. It can handle multiple genomes and uses reference genomes for scaffolding.
abaqus	Expanse	Abaqus is a software suite for finite element analysis and computer-aided engineering, primarily used in engineering and design fields for simulating the behavior of structures and materials under various conditions. It offers advanced capabilities for modeling, analysis, and visualization, making it a powerful tool for solving complex engineering challenges.	Abaqus is a software suite used for finite element analysis and computer-aided engineering simulations for a wide range of industrial applications. It provides powerful simulation capabilities to analyze the behavior of materials and structures under different conditions.

<https://access-sds.ccs.uky.edu:8080/>



ACCESS Resource Advisor

- <https://access-ara.ccs.uky.edu:8080/>
- Tool designed to help researchers (in particular new users) find appropriate infrastructure for their research.
- The term “supercomputer” refers to any substantial computation system or computing resource such as a computing cluster, cloud platform, large server system or high-performance computer.

Recommendations:

[Report Issue](#)

Expanse

Stampede-3

ACES

[See More](#) [Close](#)



Environment Modules – discover software on systems

- Expanse uses *Lmod*, a Lua based module system.
 - https://lmod.readthedocs.io/en/latest/010_user.html
- What modules let you do:
 - Dynamic modification of your shell environment
 - Choose between different versions of the same software or different combinations of related codes.
 - Setup custom environments by loading available modules into the shell environment including needed compilers and libraries and the batch scheduler.
- You can only see the software associated with the module, even though there are many other choices
- Use the command "module spider" option to see if a particular package exists and can be loaded, run command
 - module spider <package>
 - module keywords <term>
- For additional details, and to identify module dependencies modules, use the command: module spider <application_name>
- The module paths are different for the CPU and GPU nodes. Users can enable the paths by loading the CPU or GPU modules

Module Command Examples

```
[username@login02 ~]$ module reset
[username@login02 ~]$ module list
Currently Loaded Modules:
 1) cpu/0.15.4  2) slurm/expanse/20.02.3  3) intel/19.1.1.217
username@login02 ~]$ module avail
----- /cm/shared/apps/spack/cpu/lmod/linux-centos8-x86_64/intel/19.1.1.217 -----
bamtools/2.5.1      grace/5.1.25          libpng/1.6.37        openmpi/4.0.4 (D)
bedtools2/2.27.1    gsl/2.5              libtirpc/1.2.6       papi/6.0.0.1
  [SNIP]
eigen/3.3.7        jasper/2.0.16         openmpi/3.1.6
----- /cm/shared/apps/spack/cpu/lmod/linux-centos8-x86_64/Core -----
abaqus/2018          emboss/6.6.0          gmp/6.1.2          parallel/20200822
anaconda3/2020.11   freesurfer/7.1.1       go/1.15.1          pciutils/3.7.0
  [SNIP]
cmake/3.18.2          gcc/9.2.0           mpfr/4.0.2          zstd/1.4.5
curl/7.72.0          gcc/10.2.0          (D)               nbo/7.0-openblas
doxygen/1.8.17        gh/1.13.1          openjdk/11.0.2
----- /cm/local/modulefiles -----
boost/1.71.0        cmjob     lua/5.3.5      shared   singularitypro/3.5   slurm/expanse/20.02.3 (L)
----- /cm/shared/apps/xsede/modulefiles -----
cue-login-env      xdinfo/1.5-1      xdusage/2.1-1
----- /usr/share/modulefiles -----
DefaultModules      cpu/0.15.4 (L)    gct/6.2      globus/6.0      gpu/0.15.4      nostack/0.15.4
----- /cm/shared/modulefiles -----
AMDuProf/3.4.475    default-environment    sdsc/1.0
```

List Current environment

Show available modules

Where:

L: Module is loaded

D: Default Module



Hands On: Discovering Available Software and Tools

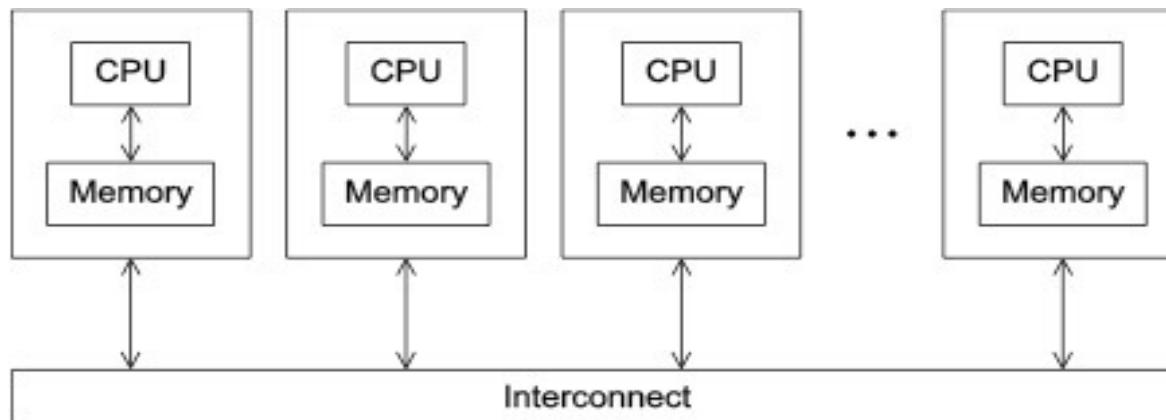
- Log onto ACCESS site
 - Tour site / links
- Check out what modules you have access to,
 - module list
- HPC system user guides
 - User guides: e.g. expanse.sdsc.edu
- Search for event information
 - SDSC Training Catalog
 - SDSC GitHub, etc.



3. Submitting Jobs to an HPC System



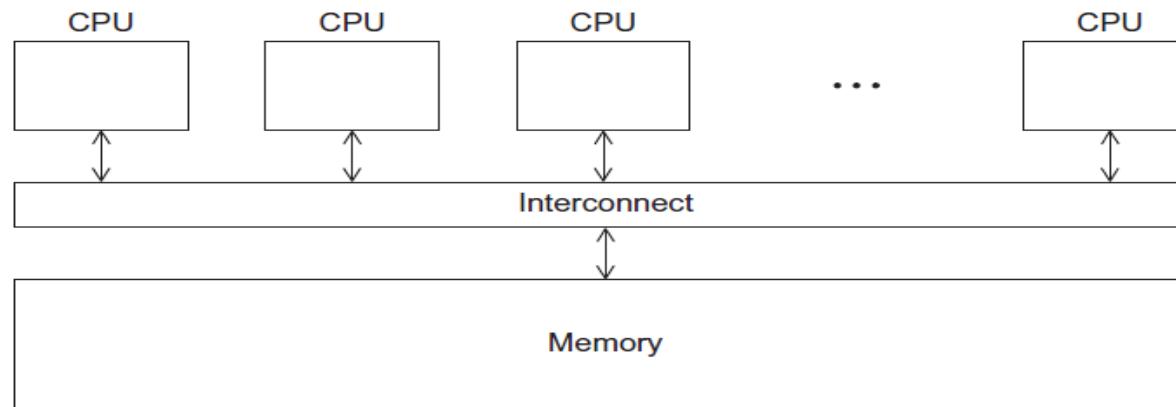
Parallel Models: Distributed Memory



- Programs run asynchronously, pass messages for communication and coordination between resources.
- Examples include: SOA-based systems, massively multiplayer online games, peer-to-peer apps.
- Different types of implementations for the message passing mechanism: HTTP, RPC-like connectors, message queues
- HPC historically uses the Message Passing Interface (MPI)



Parallel Models: Shared Memory



- CPUs all share same localized memory (SHMEM): Coordination and communication between tasks via interprocessor communication (IPC) or virtual memory mappings.
- May use: uniform/non-uniform memory access (UMA or NUMA); cache-only memory architecture (COMA).
- Most common HPC API's for using SHMEM:
 - Portable Operating System Interface (POSIX); Open Multi-Processing (OpenMP) designed for parallel computing – best for multi-core computing.

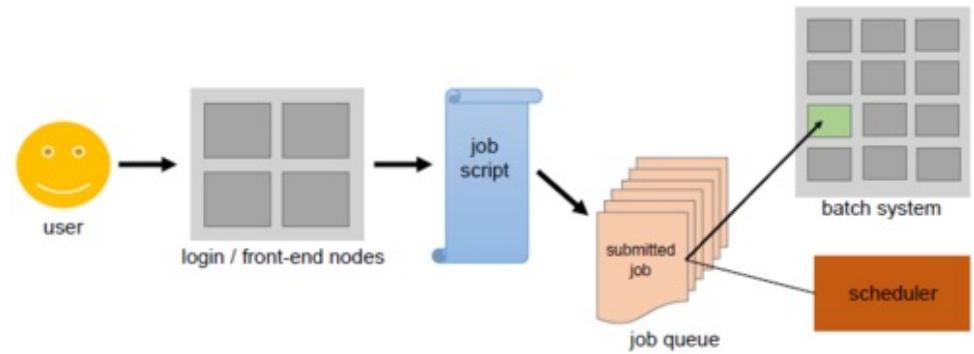


Jobs Can be Run **Interactively** or as **Batch Jobs** (Background)

Interactive Jobs:

- User has direct access one or more HPC nodes (or cores).
- Only user's job and user are allowed on the node.
- Parallel jobs and applications allowed to run on node.

The screenshot shows a Jupyter Notebook environment with several open cells. One cell displays Python code for GPU memory usage, and another shows a command-line error message about an NVIDIA driver issue. The sidebar shows pinned applications like MATLAB, SDSC, and various HPC-related tools. A file browser window is also visible.



Batch Jobs:

- Batch queue system runs on HPC system.
- Users do not run calculations interactively -- instead they submit non-interactive batch jobs to the scheduler.
- Scheduler manages when and where job will run, tracks jobs, and collects results (based on job).



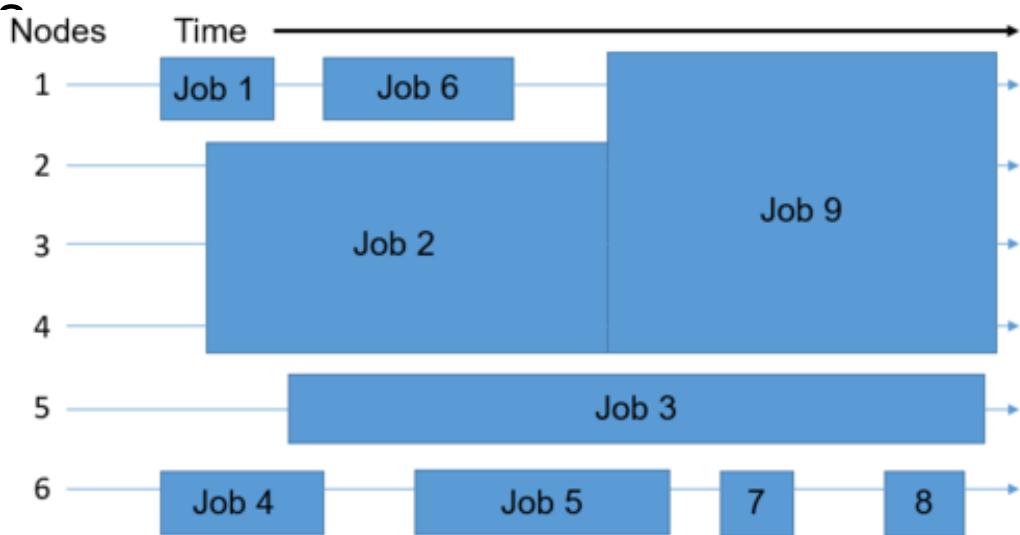
What is a Scheduler & How are They Used?/

- Any HPC (or HTC) system needs a means of sharing computational resources fairly between users.
- Batch-queueing systems are designed to do this.
- All batch systems have at least these features:
 - Scheduler for allocating resources (CPUs!) to jobs and for prioritising jobs;
 - One or more queues to which jobs are submitted.
 - Queues might be configured for a particular type of job: serial or parallel; long or short; requiring particularly high memory. These are called partitions (or job queues)
- Main goals:
 - Minimize time between job submission and completion: No job should stay in queue for extensive periods of time.
 - Optimize CPU/GPU utilization: Minimize CPU/BPU idle times.
 - Maximize job throughput: Manage as many jobs per time unit as possible.
 - Support running jobs automatically in the background



How a Scheduler Schedules Jobs

- Simple example:
 - a 6-node system
 - user wants to run 9 jobs.
- Scheduler places the jobs in the queue and then onto the available nodes as they open up.
- *Many parameters affect scheduling: number of jobs submitted, required runtime, required number of cores, required main memory, accelerators, libraries, etc.*

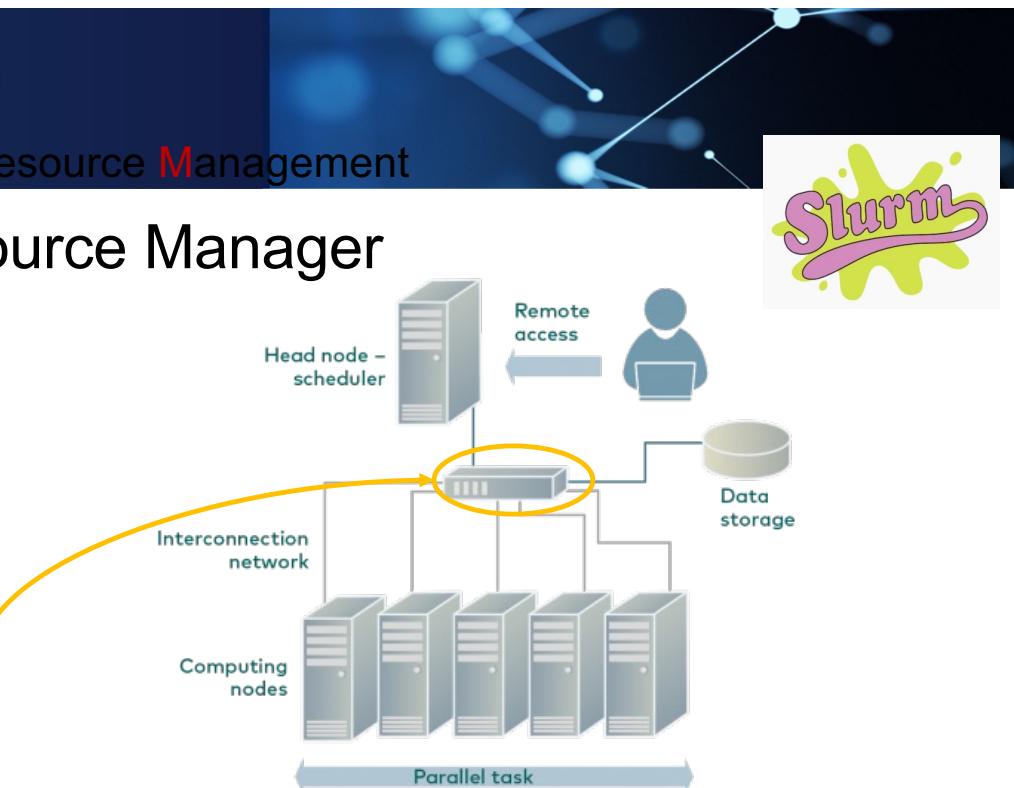
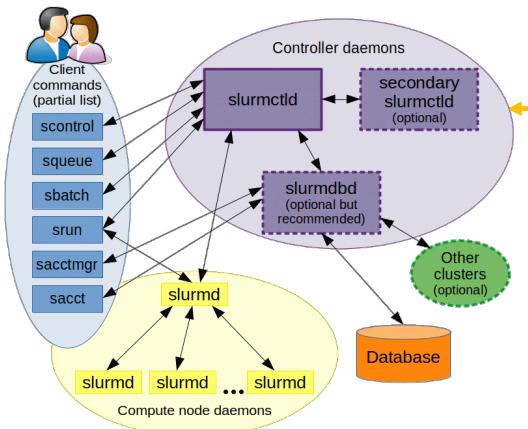


Scheduler needs to play kind of "multidimensional tetris" to fill the cluster's nodes evenly and efficiently.



Expanse Batch Jobs: Slurm Resource Manager

- Open Source, runs on many systems
- “Glue” for parallel computer to schedule and execute jobs
- Role: Allocate resources within a cluster
 - Nodes (unique IP address)
 - Interconnect/switches
 - Generic resources (e.g. GPUs)
 - Launch and otherwise manage jobs



- Functionality:
 - Prioritize queue(s) of jobs;
 - Decide when and where to start jobs;
 - Terminate job when done;
 - Appropriately allocate resources;
 - Manage accounts for jobs



Slurm Partitions on Expanse

Partition Name	Max Walltime	Max Nodes/Job	Max Running Jobs	Max Running + Queued Jobs	Charge Factor	Notes
compute	48 hrs	32	32	64	1	Used for exclusive access to regular compute nodes; <i>limit applies per group</i>
shared	48 hrs	1	4096	4096	1	Single-node jobs using fewer than 128 cores
gpu	48 hrs	4	4	8 (32 Tres GPU)	1	Used for exclusive access to the GPU nodes
gpu-shared	48 hrs	1	24	24 (24 Tres GPU)	1	Single-node job using fewer than 4 GPUs
large-shared	48 hrs	1	1	4	1	Single-node jobs using large memory up to 2 TB (minimum memory required 256G)
debug	30 min	2	1	2	1	Priority access to shared nodes set aside for testing of jobs with short walltime and limited resources
gpu-debug	30 min	2	1	2	1	Priority access to gpu-shared nodes set aside for testing of jobs with short walltime and limited resources; <i>max two gpus per job</i>
preempt	7 days	32		128	.8	Non-refundable discounted jobs to run on free nodes that can be pre-empted by jobs submitted to any other queue
gpu-preempt	7 days	1		24 (24 Tres GPU)	.8	Non-refundable discounted jobs to run on unallocated nodes that can be pre-empted by higher priority queues

https://www.sdsc.edu/support/user_guides/expanse.html#running



Batch Scripts -- Used to Launch Jobs

- **Batch Jobs:** Submit batch scripts from the login nodes to a batch service:
 - Expanse uses the Simple Linux Utility for Resource Management (SLURM) batch environment.
- Can be used to run serial jobs (1 core), multi-threaded (OpenMP), multi-core jobs, and parallel (multi-node, MPI) jobs
- Parameters you can set:
 - Partition (queue)
 - Time limit for the run (maximum of 48 hours)
 - Number of nodes, tasks per node; Memory requirements (if applicable)
 - You can define the job name, output file location; email info, configuration



https://www.sdsc.edu/support/user_guides/expanse.html#running



Hands-on: Simple Batch Script Job: ENV_Info

```
[uswername@login02 env_info ]$ cat env-slurm.sb
#!/bin/bash
#SBATCH --job-name=" env_info "
#SBATCH --output="mpi_prime.%j.%N.out"
####SBATCH --partition=compute
#SBATCH --partition=debug
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --export=ALL
#SBATCH -t 00:10:00
#SBATCH -A use300

## Environment
module purge
module load slurm
module load cpu
module load gcc/10.2.0
module load openmpi/4.0.4

## echo job name and id:
echo "SLURM_JOB_NAME: $SLURM_JOB_NAME"
echo "SLURM_JOB_ID: $SLURM_JOB_ID"
d=`date`
echo "DATE: $d"
e='env'
echo "env= $e"
```

```
[mthomas@login02 env_info ]$ sbatch env-slurm.sb
Submitted batch job 14126259
[mthomas@login02 env_info]$ !sq
squeue -u mthomas
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
14126259 debug env_info mthomas R 0:04 1 exp-9-55
[mthomas@login01 env_info ]$ cat env_info.14126259.exp-4-35.out
SLURM_JOB_NAME: env_info
SLURM_JOB_ID: 14126259
hostname= exp-4-35
date= Sun Jun 26 22:05:15 PDT 2022
whoami= mthomas
pwd= /home/mthomas/hpcctr-examples/env_info
Currently Loaded Modules: 1) slurm/expanse/21.08.8 2) cpu/0.15.4
-----
env= LD_LIBRARY_PATH=/cm/shared/apps/slurm/current/lib64/slurm:
[SNIP]
/cm/shared/apps/slurm/current/lib64
SLURM_SUBMIT_DIR=/home/mthomas/hpcctr-examples/env_info
HISTCONTROL=ignoredups
DISPLAY=localhost:16.0
HOSTNAME=exp-4-35
[SNIP]
```

Example Code: <https://github.com/sdsc-hpc-training-org/hpcctr-examples>



Slurm Commands

- sacct
- sacctmgr
- salloc
- sattach
- sbatch
- squeue
- scancel
- scontrol
- scrontab
- sdiag
- sh5util
- [sinfo](#)
- sprio
- [squeue](#)
- sreport
- srun
- sshare
- sstat
- strigger
- svview



Basic Job management

- squeue - View information about jobs in scheduling queue
- A few common commands:

-A, --account=<account_list>	Filter by accounts (comma-separated list)
j, --jobs=<job_id_list>	Filter by job IDs (comma-separated list)
-p, --partition=<partition_list>	Filter by partitions (comma-separated list)
-u, --user=<user_list>	Filter by users (comma-separated list)



SLURM Environment Variables

Internal ENV variables that exist when job is submitted:

INPUT ENVIRONMENT VARS

- Upon startup, sbatch will read and handle the options set in the following environment variables.
- `SBATCH_JOB_NAME`
 - Same as `-J`, `--job-name`
- `SBATCH_ACCOUNT`
 - Same as `-A`, `--account`
- `SBATCH_TIMELIMIT`
 - Same as `-t`, `--time`
- More...

OUTPUT ENVIRONMENT VARS

- The Slurm controller will set the following variables in the environment of the batch script.
- `SLURM_EXPORT_ENV`
 - Same as `--export`.
- `SLURM_JOB_ID`
 - The ID of the job allocation.
- `SLURM_JOB_NAME`
 - Name of the job.
- More...



Common Slurm Command Examples

- Submit jobs using the sbatch command:

```
$ sbatch mycode-slurm.sb
Submitted batch job 8718049
```

- Check job status using the squeue command:

```
$ squeue -u $USER
      JOBID PARTITION      NAME      USER      ST          TIME   NODES NODELIST(REASON)
      8718049    compute      mycode    user      PD          0:00        1      (Priority)
```

- Once the job is running, monitor its state:

```
$ squeue -u $USER
      JOBID PARTITION      NAME      USER      ST          TIME   NODES NODELIST(REASON)
      8718049     debug      mycode    user      R           0:02        1      expanse-14-01
```

- Cancel a running job:

```
$ scancel 8718049
```



SLURM “srun” Command

- Used to launch a parallel job on cluster managed by Slurm.
- If necessary, srun will first create a resource allocation in which to run the parallel job.
- Common arguments used on Expanse:
 - `--mpi=<mpi_type>` Identify the type of MPI to be used. Use ‘pmi2’
 - `-n, --ntasks=<number>` Specify the number of tasks to run.
 - `-cpu-bind`: bind tasks to CPUs
- what is the difference between mpirun and SLURM srun?
 - srun is optimized for Expanse (via the PMI interface) and more efficiently allocates, organizes, and starts up the MPI processes.

```
srun --mpi=pmi2 -n 24 --cpu-bind=rank ./mpi_prime Y
```

<https://slurm.schedmd.com/srun.html>



HPC Systems Have Multiple File Systems (Expanse)

- Home directories (/home/\$USER) – 8 week rolling backup.
 - Login nodes: 100GB limit;
 - Use for source trees, binaries, and small input files.
 - Not good for large scale I/O.
- Lustre filesystems: Good for scalable large block I/O
 - /expanse/lustre/scratch/\$USER/temp_project
 - 2.5PB; peak performance: 100GB/s. Good for storing large scale scratch data during a job.
 - /expanse/lustre/projects/-
 - 2.5PB, peak performance: 100 GB/s. Long term storage.
 - Not good for large # of small files or small block I/O
- Local node “scratch” (SSD) filesystems
 - /scratch local to each native compute node – 210GB on regular compute nodes, 285GB on GPU, large memory nodes, 1.4TB on selected compute nodes.
 - SSD location is good for writing small files and temporary scratch files. Purged at the end of a job.
- Webinar on Data Management and File Systems:
 - https://www.sdsc.edu/event_items/202110_ExpanseWebinar-M.Shantharam.html



Hands On: Logging on and checking account info

```
[mthomas@login02 ~]$ expanse-client user
Resource expanse
```

CIS250186: AI Unlocked: Empowering Higher Education through Research and Discovery					Active
Discover: Feb 26, 2025 to Feb 25, 2026					
Overview	Credits + Resources	Users + Roles	History		
Resource	Status	Balance	End Date	My Username	
Delta CPU	Active	15K of 15K Core-hours remaining (100%)	Feb 25, 2026	thomasm	
NCSA Delta GPU	Active	5K of 5K GPU Hours remaining (100%)	Feb 25, 2026	thomasm	
NCSA DeltaAI	Active	5K of 5K GPU Hours remaining (100%)	Feb 25, 2026	thomasm	
SDSC Expanse CPU	Active	15K of 15K Core-hours remaining (100%)	Feb 25, 2026	mthomas	
SDSC Expanse GPU	Active	8K of 8K GPU Hours remaining (100%)	Feb 25, 2026	ux415294	

	NAME	STATE	PROJECT	TG PROJECT	USED	AVAILABLE	USED BY PROJECT
..... [SNIP]							
3	mthomas	allow	abc123	TG-abcd	0	40000	15091
4	mthomas	allow	ukl119	TG-CIS250186	0	15000	0
5	mthomas	allow	abc123		2908	5050000	4427508

```
[mthomas@login02 ~]$
```



Hands On: Submitting a Job to the SLURM Batch Queue

```
[mthomas@login01 ENV_INFO]$ cat env-slurm.sb
#!/bin/bash
#SBATCH --job-name=envinfo
#SBATCH --output="envinfo.%j.%N.out"
#SBATCH --partition=compute
#SBATCH --nodes=1
#SBATCH --tasks-per-node=1
#SBATCH --export=ALL
#SBATCH -t 00:01:00

## Environment
module purge
module load slurm
module load cpu

## perform some basic unix commands
echo "-----"
echo "hostname= `hostname`"
echo "date= `date`"
echo "whoami= `whoami`"
echo "pwd= `pwd`"
echo "module list= `module list`"
echo "-----"
echo "env= `env`"
echo "-----"
```

```
[mthomas@login01 ENV_INFO]$ cat envinfo.108867.exp-6-56.out
-----
hostname= exp-6-56
date= Wed Oct 7 23:45:43 PDT 2020
whoami= mthomas
pwd= /home/mthomas/DEMO/ENV_INFO
Currently Loaded Modules:
 1) slurm/expanse/20.02.3 2) cpu/1.0
-----
env= SLURM_MEM_PER_CPU=1024
LD_LIBRARY_PATH=/cm/shared/apps/slurm/current/lib64/slurm:/cm/shared/apps/slurm/current/lib64 LS_COLORS=rs=0

[SNIP]

MODULESHOME=/usr/share/lmod/lmod LMOD_SETTARG_FULL_SUPPORT=no
HISTSIZE=5000 LMOD_PKG=/usr/share/lmod/lmod
LMOD_CMD=/usr/share/lmod/lmod/libexec/lmod SLURM_LOCALID=0
LESSOPEN=||/usr/bin/lesspipe.sh %s LMOD_FULL_SETTARG_SUPPORT=no
LMOD_DIR=/usr/share/lmod/lmod/libexec BASH_FUNC_module%%=() { eval
$(${LMOD_CMD} bash "$@") && eval $((${LMOD_SETTARG_CMD}:--s sh) }
BASH_FUNC_ml%%=() { eval ${${LMOD_DIR}/ml_cmd "$@") } _=/usr/bin/env
-----
```



Hands On: Running “MPI-Prime” – README FILE

Many example codes in training repo: <https://github.com/sdsc-hpc-training-org/hpctr-examples>

```
[mthomas@login02]$ cat README.txt
[1] Compile:
module purge
module load slurm
module load cpu
module load gcc/10.2.0
module load openmpi/4.1.1

mpicc -o mpi_prime mpi_prime.c

[2] Run as batch job:
sbatch mpi-prime-slurm.sb

[3] Run using an interactive node:
Method [3a]
Request the interactive node using the "srun" command:

srun --partition=debug --pty --account=use300 --nodes=1 --ntasks-per-node=24 --mem=8G
-t 00:30:00 --wait=0 --export=ALL /bin/bash

Run the code using mpirun:
mpirun -n 64 ./mpi_prime 5000000
```

Codes in the repo have
README files to help you
get started



Hands On: Running “Calc Prime” -- Compiling

```
[mthomas@login02]$ pwd
/home/mthomas/sdsc.git/hpctr-examples/calc-prime
[mthomas@login02]$ module purge
[mthomas@login02]$ module load slurm cpu gcc/10.2.0 openmpi/4.1.1
[mthomas@login02]$ module list
Currently Loaded Modules:
 1) slurm/expanse/23.02.7  2) cpu/0.17.3b (c)  3) gcc/10.2.0/npcyll4
 4) ucx/1.10.1/dnpjjuc  5) openmpi/4.1.1/ygduf2r
[mthomas@login02]$ mpicc -o mpi_prime mpi_prime.c
[mthomas@login02]$ ll
total 106
-rwxr-xr-x  1 mthomas use300 22816 Mar 31 23:19 mpi_prime
-rw-r--r--  1 mthomas use300  1110 Mar 24 17:50 mpi_prime.37713064.exp-9-55.out
-rw-r--r--  1 mthomas use300   5194 Oct 11  2023 mpi_prime.c
-rw-r--r--  1 mthomas use300    874 Mar 19 23:06 mpi-prime-slurm.sb
-rw-r--r--  1 mthomas use300  1441 Oct 11  2023 README.txt
```



MPI Hello MPI: Batch Script

- To run the job, use the **sbatch** script submission command.
- Monitor the job until it is finished using the **squeue** command.
- Be sure to compile your code: see instructions in the github repo
- Script submitting to CPU, 2 nodes 128 tasks/node

```
#!/bin/bash
#SBATCH --job-name="mpi_prime"
#SBATCH --output="mpi_prime.%j.%N.out"
##SBATCH --partition=compute
#SBATCH --partition=debug
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=24
#SBATCH --mem=8GB
#SBATCH --export=ALL
#SBATCH -t 00:10:00
#SBATCH -A use300

module purge
module load slurm
module load cpu
module load gcc/10.2.0
module load openmpi/4.1.1

## echo job name and id:
echo "SLURM_JOB_NAME: $SLURM_JOB_NAME"
echo "SLURM_JOB_ID: $SLURM_JOB_ID"
d=`date`
echo "DATE: $d"
NHI=$1
echo "Var NHI: $NHI"

## Use srun to run the job, pass variable to code
## default N_HI=262144
srun --mpi=pmi2 -n 2 ./mpi_prime 300000
```

```
[mthomas@login02]$ cat mpi_prime.37713064.exp-9-55.out
SLURM_JOB_NAME: mpi_prime
SLURM_JOB_ID: 37713064
DATE: Mon Mar 24 17:50:17 PDT 2025
Var NHI:
The argument supplied is 300000
24 March 2025 05:50:18 PM
PRIME_MPI
n_hi= 300000 C/MPI version
An MPI example program to count the number of primes.
The number of processes is 2
      N          Pi        Time
      1            0   0.000018
      2            1   0.000001
      4            2   0.000000
[SNIP]
      1024         172   0.000356
      2048         309   0.001269
      4096         564   0.004654
      8192        1028   0.016893
      16384        1900   0.062876
      32768        3512   0.232420
      65536        6542   0.869716
      131072       12251   3.270207
      262144       23000  12.307361

PRIME_MPI - Master process:
Normal end of execution.
24 March 2025 05:50:34 PM
```



Hands On: Submitting “Hello CUDA” GPU Job

```
/*
 * Copyright 1993-2010 NVIDIA Corporation.
 * All rights reserved.
 * NVIDIA Corporation and its licensors
 * retain all intellectual property and
 *
 * Updated by Mary Thomas, April 2023, for
 * simple cuda compile example
 */
#include <stdio.h>
__global__ void kernel( void ) { }
int main( void ) { kernel<<<1,1>>>();  
printf( "Hello, SDSC HPC Training World!\n"  
); return 0;  
}
```

- To run the job, use the **sbatch** script submission command on the login node.
- Monitor the job until it is finished using the **squeue** command.
- Be sure to compile your code: see instructions in the github repo README file.

```
[mthomas@login02]$ cat hello-cuda.sb
#!/bin/bash
#SBATCH --job-name="hello_cuda"
#SBATCH --output="hello_ cuda.%j.%N.out"
#SBATCH --partition=gpu-shared
#SBATCH --nodes=1
#SBATCH --gpus=1
#SBATCH --ntasks-per-node=1
#SBATCH --cpus-per-task=1
#SBATCH --mem=93G
#SBATCH --account=use300
#SBATCH --no-requeue
#SBATCH -t 00:05:00
module purge
module load gpu
module load slurm
./hello_cuda
```

```
[mthomas@login02]$ squeue -u mthomas
      JOBID PARTITION      NAME      USER ST      TIME  NODES NODELIST(REASON)
      38028463  gpu-share  hello_cu  mthomas PD      0:00      1 (Priority)

[mthomas@login02]$ cat hello_ cuda.38028528.exp-1-59.out
Hello, SDSC HPC Training World!
```

NAIRR Pilot

National Artificial Intelligence
Research Resource Pilot



Interactive HPC Access



Interactive HPC Methods & Applications

- In computer science, [interactive computing](#) refers to software which accepts input from the user as it runs.
- Interact with data after job is done:
 - Unix: query file info, location, output, grep, awk, sed
 - Cat the file contents from batch job or raw data
 - NetCDF data browser
- Plot results:
 - From within the code/model using libraries
 - Command-line driven graphing utility : Gnuplot
- Data visualization apps
 - NetCDF, HPF, TAU, ParaView
 - other
- Data Analysis Platforms: Matlab, R



Interactive HPC Computing- Motivation

- Need more memory: Your jobs no longer fit onto the CPU/system you have been using:
 - That is why Expanse has 768 nodes (128 cores per node), with 256 GB DDR memory on each node
 - My MacBook Pro: 1 node, 8 cores, 16 GB DDR
- Too much data: your application needs more room:
 - Expanse has 1TB NVME/node, and 12 PB file system.
 - My MBP: 500 GB, no NVME
- Your network is too slow:
 - Expanse has connections to ~ 150GB/sec (or faster) networks
 - My MBP: 300 Mbps download; 11 Mbps upload
 - Bioinfo lab, running analysis on PC: FastQ dataset ~ 500 MB: would need 360+ seconds to upload 1 run.



Accessing Interactive Compute Nodes on Expanse

- Connect to HPC system (e.g. Expanse) via terminal using SSH □ secure connections
- Use the srun command to obtain nodes for ‘live,’ command line interactive access:

CPU	srun --partition=debug --pty --account= abc123 --nodes=1 --ntasks-per-node=4 --mem=8G -t 00:30:00 --wait=0 --export=ALL /bin/bash
GPU	srun --partition=gpu-debug --pty --account= abc123 --ntasks-per-node=10 --nodes=1 --mem=96G --gpus=1 -t 00:30:00 --wait=0 --export=ALL /bin/bash

- (Tested March/2025)



Using An Interactive CPU node

```
[mthomas@login01 calc-prime]$ srun --partition=debug --pty --account=abc123 --  
nodes=1 --ntasks-per-node=16 --mem=8G -t 00:30:00 --wait=0 --export=ALL /bin/bash
```

```
srun: job 24457429 has been allocated resources  
[mthomas@exp-9-55 calc-prime]$ module purge  
[mthomas@exp-9-55 calc-prime]$ module load slurm  
[mthomas@exp-9-55 calc-prime]$ module load cpu  
[mthomas@exp-9-55 calc-prime]$ module load gcc/10.2.0  
[mthomas@exp-9-55 calc-prime]$ module load openmpi/4.1.1  
[mthomas@exp-9-55 calc-prime]$ mpirun -n 16 ./mpi_prime  
06 August 2023 11:10:26 PM  
PRIME_MPI n_hi= 5000000 C/MPI version  
An MPI example program to count the number of primes: # processes is 64  
      N      Pi      Time  
      1      0     0.013258  
      2      1     0.001058  
      4      2     0.000101  
      8      4     0.000101  
[SNIP]  
 131072    12251    0.110848  
 262144    23000    0.410792  
 524288    43390    1.527210  
1048576    82025    5.733612  
2097152   155611   21.725862  
PRIME_MPI - Master process: Normal end of execution.  
06 August 2023 11:12:26 PM
```

Request an interactive node
for 30 minutes

- Beware of oversubscribing your job: don't ask for more cores than you have requested.
- Intel compiler allows this, but your performance will be degraded.
- Exit interactive session when your work is done or you will be charged more CPU time.



Using Interactive GPU nodes

```
[snip]  
Last login: Fri Feb 18 12:58:32 2022 from 76.176.117.51
```

```
[username@login02 ~]$
```

```
[username@login02 ~]$ srun --partition gpu-debug --pty --account=use300 --ntasks-per-node=10 --nodes=1 --mem=96G  
--gpus=1 -t 00:30:00 --wait=0 --export=ALL /bin/bash
```

```
srun: job 9794018 queued and waiting for resources
```

```
srun: job 9794018 has been allocated resources
```

```
[mthomas@exp-14-57 ~]$
```

```
[mthomas@exp-14-57 ~]$ nvidia-smi
```

```
Fri Feb 18 13:04:19 2022
```

```
+-----+  
| NVIDIA-SMI 460.32.03 Driver Version: 460.32.03 CUDA Version: 11.2 |  
+-----+  
| GPU Name Persistence-M| Bus-Id Disp.A | Volatile Uncorr. ECC | |
| Fan Temp Perf Pwr:Usage/Cap| Memory-Usage | GPU-Util Compute M. |  
| | | | MIG M. |  
+=====+=====+=====+  
| 0 Tesla V100-SXM2... On | 00000000:86:00.0 Off | 0 |  
| N/A 34C P0 41W / 300W | 0MiB / 32510MiB | 0% Default |  
| | | N/A |  
+-----+-----+-----+  
  
+-----+  
| Processes: |  
| GPU GI CI PID Type Process name GPU Memory |  
| ID ID Usage |  
+=====+  
| No running processes found  
+-----+  
[username@login02 ~]$ exit
```

Request an interactive node
for 30 minutes

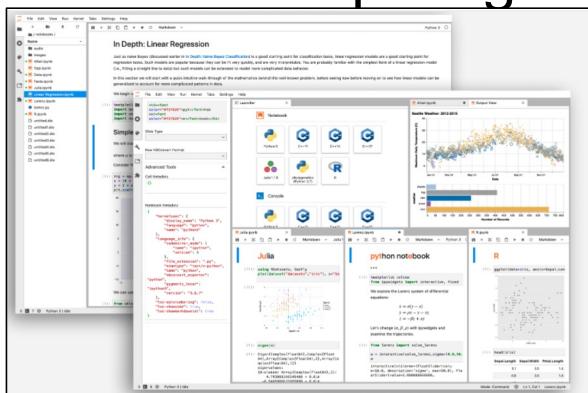
Verify you are on a GPU node

Exit when tasks are done



Interactive HPC Computing

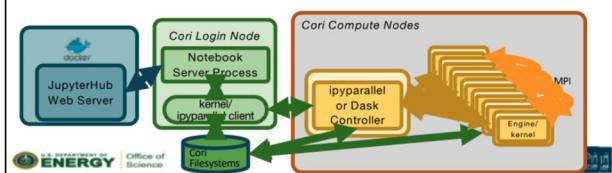
<https://jupyter.org/>



Interactive Distributed Computing with Jupyter (NERSC)

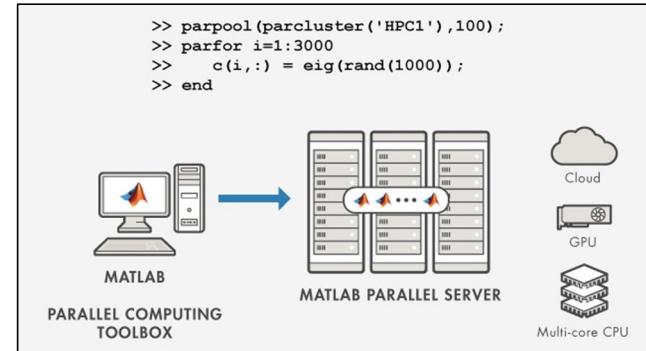
Jupyter architecture

- Allocate nodes on Cori interactive queue and start ipyparallel or Dask cluster
 - Developed %ipcluster magic to setup within notebook
- Compute nodes traditionally do not have external address
 - Required network configuration / policy decisions
- Distributed training communication is via MPI Horovod or Cray ML Plugin



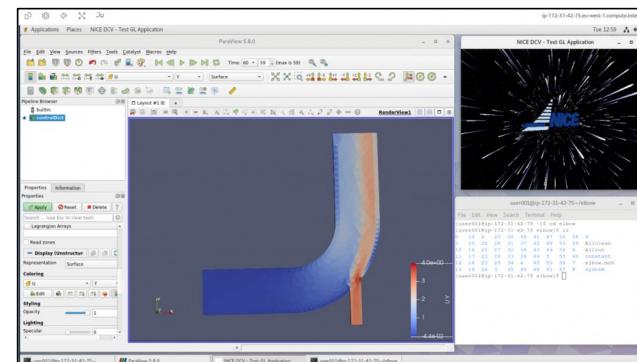
<https://drive.google.com/file/d/1-OFjrk1q3L1d3uakr2xkozrPn2c2VZpZ/view>

```
>> parpool(parcluster('HPC1'),100);
>> parfor i=1:3000
>> c(i,:) = eig(rand(1000));
>> end
```



Parallel Matlab
(hosted on AWS)

<https://azuremarketplace.microsoft.com/en-us/marketplace/apps/mathworks-inc.matlab-parallel-server-listing?tab=Overview>



Paraview
(running on AWS)

<https://aws.amazon.com/blogs/compute/how-to-run-3d-interactive-applications-with-nice-dcv-in-aws-batch/>

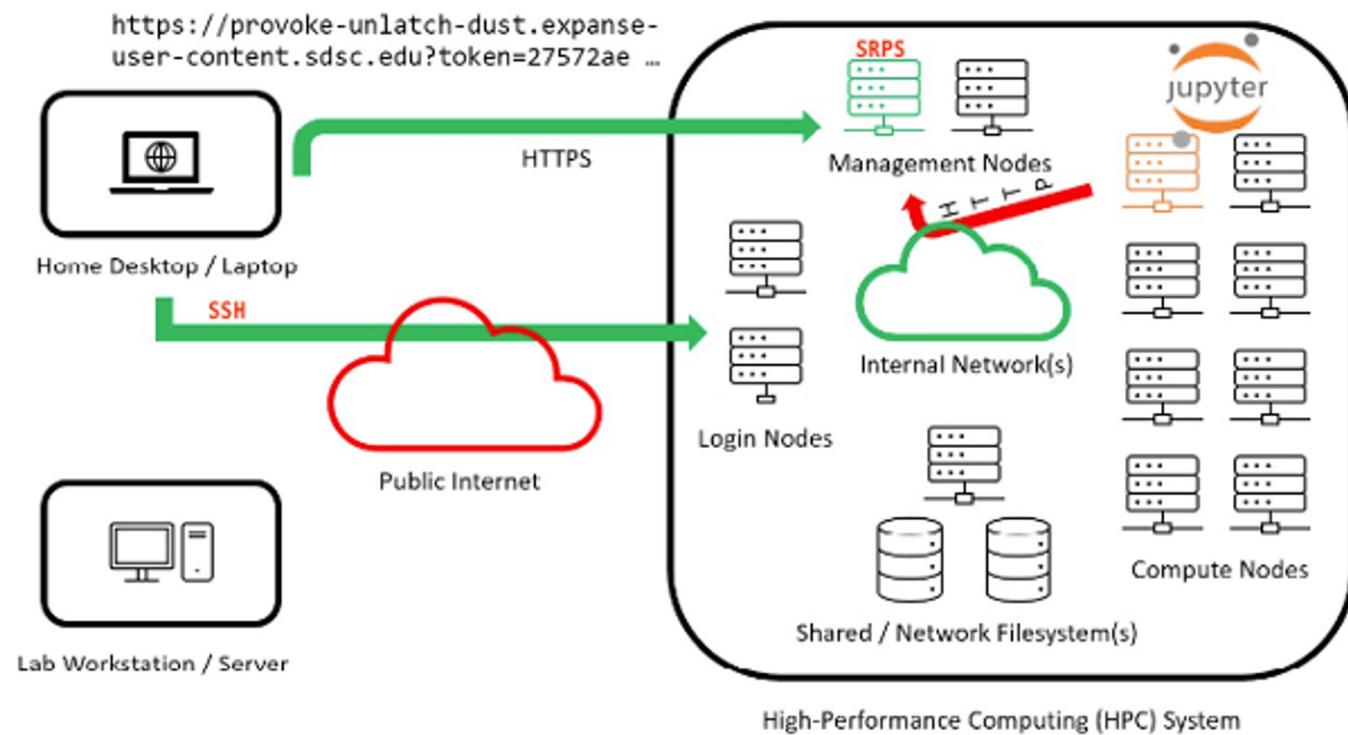


Runing Notebooks securely using *galyleo*

- 2nd generation shell utility developed to orchestrate a user's interaction with both Satellite and Slurm to start a Jupyter session within a batch job.
- Developed while reviewing start-jupyter (prototype client) codebase to sort out how best to support Expanse (OOD) Portal and HPC User Services Group long-term; integrated into an existing SSH tunneling orchestration utility to use Satellite proxy service instead
- Key features in design:
 - HTTPS URL
 - Supports containers (Singularity on Expanse)
 - GPU environment
 - No need to install conda environment or update packages
 - Increases flexibility for users to configure software environment; but also try to makes it simpler for them to do this themselves
 - Batch job script is generated completely on-the-fly.
 - Command-line argument driven.
 - Quiet mode for OOD portal

<https://github.com/sdsc/galyleo>

60

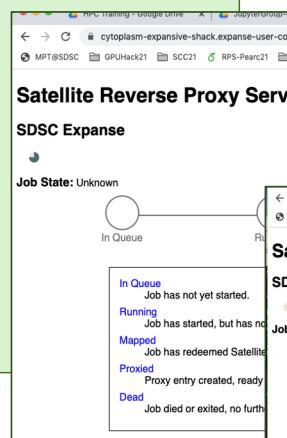


Src: M Kandes: https://education.sdsc.edu/training/interactive/202112_running_jupyter_notebooks_on_expanse/



Secure Notebooks

```
[username@login01 ~]$ export PATH="/cm/shared/apps/sdsc/galileo:${PATH}"
[username@login01 ~]$ galileo.sh --help
USAGE: galileo.sh launch [command-line option] {value}
command-line option : value
-A | --account :
-R | --reservation :
-p | --partition :
-q | --qos :
-N | --nodes :
-n | --tasks-per-node :
-c | --cpus-per-task :
-M | --memory-per-node : GB
-m | --memory-per-cpu : GB
-G | --gpus :
| --gres :
-t | --time-limit :
-j | --jupyter :
-d | --notebook-dir :
-r | --reverse-proxy :
-D | --dns-domain :
-s | --sif :
-B | --bind :
| --nv :
-e | --env-modules :
| --conda-env :
-Q | --quiet :
```



carload-spray-koala.expanse-user-content.sdsc.edu/lab

File Edit View Run Kernel Tabs Settings Help

numpy_intro.ipynb x hello_world_gpu.ipynb x hello_world_cpu.ipynb x boring_python_chap X

/ notebook-examples / Hello_World /

Name	Last Modified
hello_world_... (green)	7 months ago
hello_world_... (blue)	7 months ago
hello.py	7 months ago
README.md	7 months ago

Hello World

FileName: hello_world_cpu.ipynb

CPU Version

No package dependencies

```
[8]: print('Hello world!!!!')
Hello world!!!!
```

```
[9]: # Import hello module
import hello

# Define a local function
def world2(name):
    print(name)

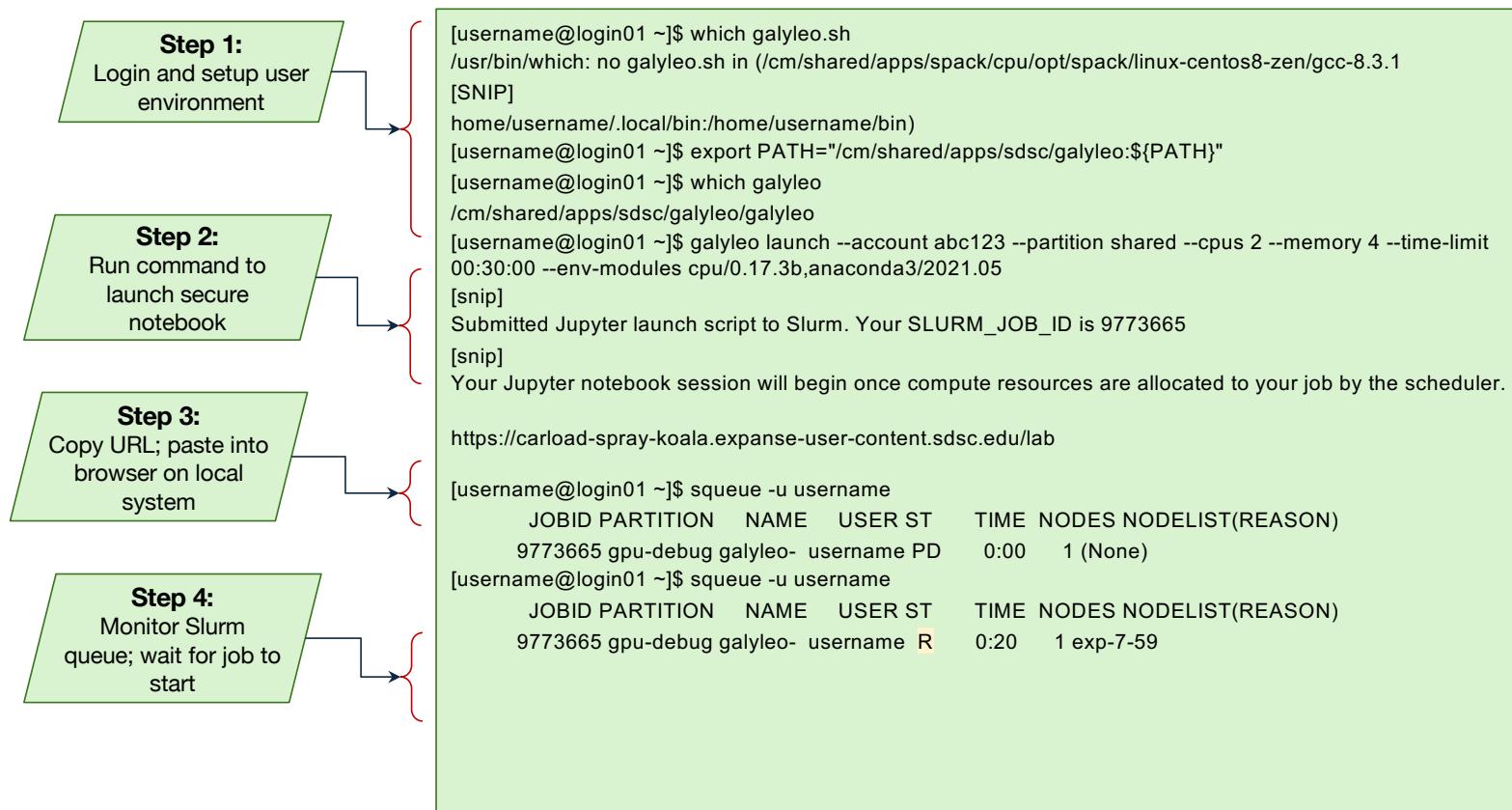
[10]: # Call function
world2("mary")
mary
```

```
[11]: hello.greeting("good times")
Greetings, good times
```

```
[12]: hello.world("World.")
Hello, World.
```



Launching Notebooks Using galileo from the Command Line



Expanse User Portal

Secure (HTTPS) environment

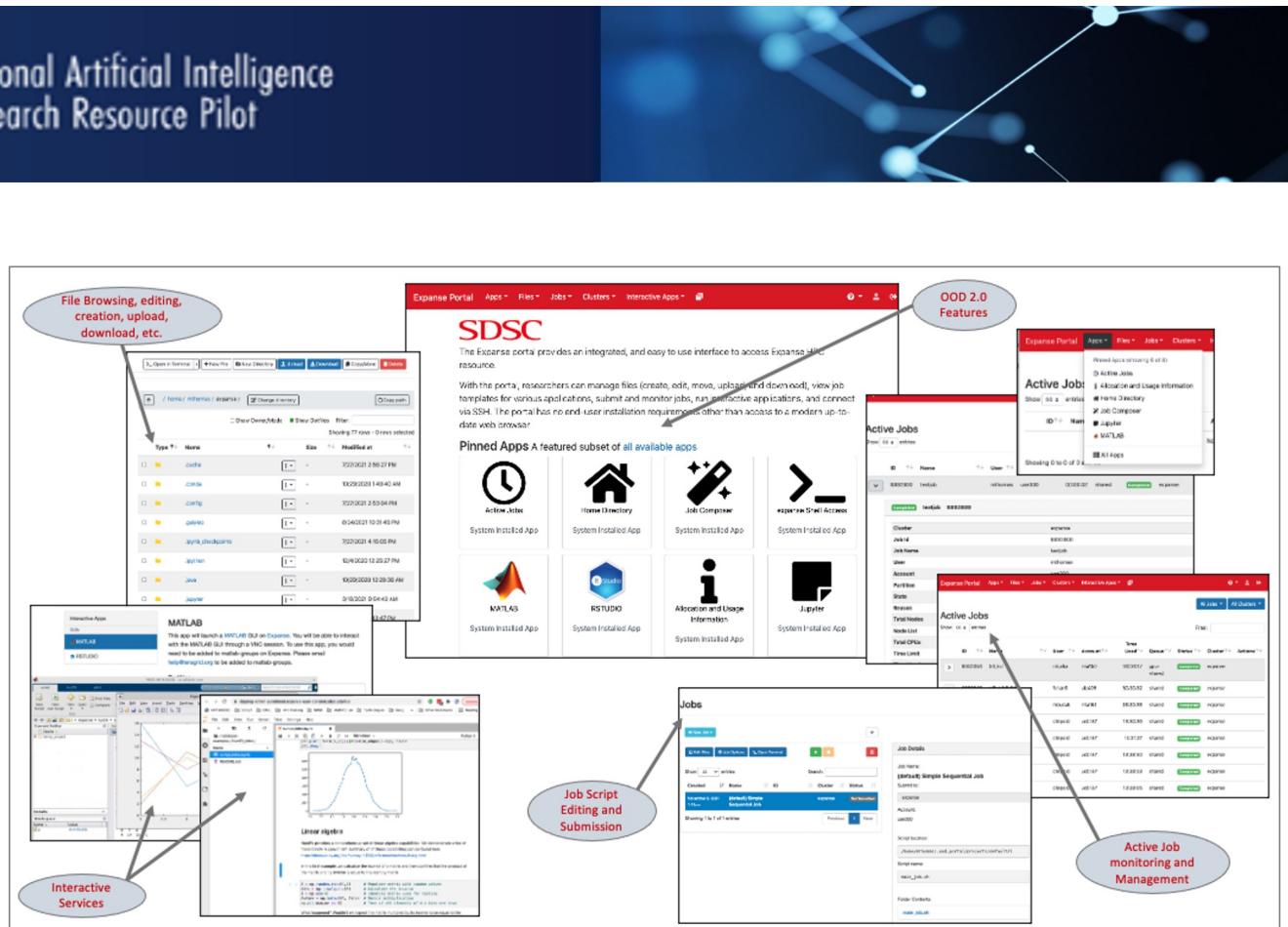
- Authenticate using ACCESS credentials
- Manages software dependencies

Secure services including:

- Data & File Management
- Job Builder
- Batch Job Submission
- Job Monitoring

Interactive applications:

- Jupyter Notebooks
- Jupyter Lab
- Matlab
- Rstudio



<https://portal.expanse.sdsc.edu>



Expanse Portal: File Management

- Can view folders and files on Expanse

Host: login.expanse.sdsc.edu Initial directory: /home/mthomas/expanse

```

-rw----- 1 mthomas use300 17803 Aug 1 21:33 .viminfo
-rw-r--r-- 1 mthomas use300 36 Jan 27 2022 .vimrc
drwx----- 2 mthomas use300 4 Aug 1 23:11 .vnc
-rw-r--r-- 1 mthomas use300 173 Oct 7 2020 .wget-hsts
-rw-r--r-- 1 mthomas use300 124 Oct 7 2020 README.txt
drwxr-xr-x 4 mthomas use300 4 Apr 15 15:37 classes
drwxr-xr-x 2 mthomas use300 6 Feb 16 18:41 comet-files
-rw-r--r-- 1 mthomas use300 116 Mar 4 2021 conda-activate.txt
drwxr-xr-x 2 mthomas use300 4 Oct 29 2020 conda-install-tmp
drwxr-xr-x 8 mthomas use300 8 Feb 16 20:39 dev
-rw-r--r-- 1 mthomas use300 12266 Nov 8 2021 ex.cl.cmds
drwxr-xr-x 3 mthomas use300 8 Jul 27 02:27 galyleo-examples
drwxr-xr-x 5 mthomas use300 9 Feb 16 18:36 galyleo-repo
drwxr-xr-x 2 mthomas use300 10 Aug 1 20:50 gpuHack-ex
drwxr-xr-x 3 mthomas use300 3 May 11 15:40 gpuHack22
drwxr-xr-x 13 mthomas use300 16 Jun 27 13:31 hpctr-examples
drwxr-xr-x 5 mthomas use300 6 Feb 16 20:47 hpctrain
drwxr-xr-x 2 mthomas use300 8 Jul 27 02:33 interactive.ex
drwxr-xr-x 2 mthomas use300 3 Aug 1 21:33 matlab-ex
drwxr-xr-x 24 mthomas use300 27 Jul 27 2021 miniconda3
drwxr-xr-x 3 mthomas use300 5 Jul 27 2021 ml-dev-mary
-rw----- 1 mthomas use300 235 Jun 1 2021 modules.cpu.txt
-rwx----- 1 mthomas use300 84 Feb 8 2021 modules.gpu.txt
-rw-r--r-- 1 mthomas use300 6178 Mar 4 2021 modules.marty.ex.txt
drwxr-xr-x 3 mthomas use300 4 Jul 18 16:09 nn-pde-TEST
drwxr-xr-x 10 mthomas use300 13 Jul 28 2021 notebook-examples
drwxr-xr-x 22 mthomas use300 26 Jul 20 2021 notebook-examples-dev
-rwxr-xr-x 1 mthomas use300 234277552 Aug 1 20:04 ocean_his.nc.gz
drwxr-xr-x 9 mthomas use300 19 Jul 28 2021 reverse-proxy
drwxr-xr-x 2 mthomas use300 7 Feb 16 18:44 scc21
drwxr-xr-x 2 mthomas use300 5 Feb 16 20:37 tensorflow
drwxr-xr-x 2 mthomas use300 3 Jul 14 2021 tools

```

[mthomas@login01 ~]\$

Type	Name	Size	Modified at
Folder	classes	-	4/15/2022 3:37:19 PM
Folder	comet-files	-	2/16/2022 6:41:00 PM
Folder	conda-install-tmp	-	10/29/2020 2:27:20 AM
Folder	dev	-	2/16/2022 8:39:49 PM
Folder	galyleo-examples	-	7/27/2022 2:27:08 AM
Folder	galyleo-repo	-	2/16/2022 6:36:21 PM
Folder	gnuplot-ex	-	8/1/2022 8:50:45 PM
Folder	gpuHack22	-	5/11/2022 3:40:13 PM
Folder	hpctr-examples	-	6/27/2022 1:31:14 PM
Folder	hpctrain	-	2/16/2022 8:47:23 PM



Secure Services: Job Builder

Secure services including:

- Job Builder
- Hands-on:
 - Modify and run hello-AI-Unlocked
 - Build a simple job: using Calc Prime Number



Expanse Portal: Running Matlab

Step 1: Fill out Matlab Launch form

MATLAB

This app will launch a **MATLAB** GUI on **Expanse**. You will be able to interact with the MATLAB GUI through a VNC session. Please email help@xsede.org to be added to matlab-groups.

Partition

compute

Reservation

Number of hours

1

Account

use300

I would like to receive an email when the session starts

Working directory

home

Number of cores

1

Memory (GB)

64

Launch

* The MATLAB session data for this session can be accessed under the [data](#) tab.

Step 2: Launch Matlab

Step 3: Monitor Matlab Job

Session was successfully deleted.

Home / My Interactive Sessions

Interactive Apps

MATLAB (14834793)

Host: >_exp-1-18.expanse.sdsc.edu

Created at: 2022-08-01 22:48:55 PDT

Time Remaining: 25 minutes

Session ID: fee7f0fb-48df-46af-8f40-ea

Compression: 0 (low) to 9 (high)

Launch MATLAB

Step 4: Launch Matlab

Figure 1

New to MATLAB? See resources for [Getting Started](#).

```
>> plot sin(2x)
Error using plot
    Invalid first data argument.

>> x = 0pi/100:2*pi;
>> y = sin(x);
>> plot(x,y);
Warning: MATLAB has disabled some advanced rendering features by switching to software rendering. For more information, click here.
>>
```

Plot of sin(2x) from 0 to 7.



Expanse Portal: Launching Jupyter Notebooks or Jupyter Lab

Step 1: Fill out Jupyter Launch form

The screenshot shows the 'Jupyter Session' configuration page. It includes fields for Account (use300), Partition (shared), Time limit (30), Number of cores (1), Memory required per node (2 GB), and GPUs (0). It also has sections for Singularity Image File Location and Conda Environment.

Open OnDemand / Jupyter Session

Jupyter Session

Account:
use300

Partition (Please choose the gpu, gpu-shared, or gpu-preempt as the partition if using gpus):
shared

Time limit (min):
30

Number of cores:
1

Memory required per node (GB):
2

GPUs (optional):
0

Singularity Image File Location: (Use your own or to include from existing container library at /cm/shared/apps/container e.g., /cm/shared/apps/containers/singularity/pytorch/pytorch-latest.sif)
/cm/shared/apps/containers/singularity/pytorch/pytorch-latest.sif

Environment modules to be loaded (E.g., to use latest version of system Anaconda3 include
cpu,gcc,anaconda3):
singularityro

Conda Environment (Enter your own conda environment if any):

File Edit View Run Kernel Tabs Settings Help

blouse-mustang-tusk.expanse-user-content.sdsc.edu/lab/tree/notebook-examples>Hello_W... Other Bookmarks

numpy_intro.ipynb hello_world_gpu.ipynb Python 3 (ipykernel)

Filter files by name

/notebook-examples /Hello_World/

Name	Last Modified
hello_worl...	a year ago
hello_worl...	24 minutes ago
hello.py	a year ago
README.md	a year ago

```
o iperf x
lrvv svm_l
ushbyasid
avic v_vm
id overflow
llc cqm_mbm_total cqm_mbm_local clzer
saveerptr wbnoinvd amd_ppin arat npt
ock nrp_save tsc_scale vmcb_clean fl
decodeassists pausefilter pfthreshold
save_vmload vgif v_spec_ctrl umip rdp
w_recov succor smca sme sev sev_es
```

[9]: # Check to see if system is GPU:
!nvidia-smi

NVIDIA-SMI has failed because it couldn't communicate with the NVIDIA driver. Make sure that the latest NVIDIA driver is installed and running.

[10]: # if you see: /bin/bash: nvidia-smi: command not found
the system is not GPU

[]:

[1]:

Open OnDemand / Jupyter Session

Jupyter Session

2022-08-02 00:13:41 https://blouse-mustang-tusk.expanse-user-content.sdsc.edu?
-0700 token=111d82c22f073ef486f91a72270e79be

2022-08-02 01:22:26 https://taste-headcount-rippling.expanse-user-content.sdsc.edu?
-0700 token=7b4cddee9b116386792d7997a4d7d5c1



Expanse Portal: Input data example for Jupyter Notebook

ELEMENT	VALUE
Account:	ukl119 (only works for this NAIRR workshop)
Partition: (choose gpu, gpu-shared, or gpu-preempt as the partition if using gpus)	debug. Or. shared
Time limit (min):	30
Number of cores	1
Memory required per node (GB)	2
GPUs (optional):	0
Singularity Image File Location: (Use your own or to include from existing container library at /cm/shared/apps/container e.g., /cm/shared/apps/containers/singularity/pytorch/pytorch-latest.sif)	/cm/shared/apps/containers/singularity/pytorch/pytorch-latest.sif OR /cm/shared/apps/containers/singularity/tensorflow/tensorflow-latest.sif
Environment modules to be loaded: (E.g., to use latest version of system Anaconda3 include cpu,gcc,anaconda3)	singularitypro
Conda Environment: (Enter your own conda environment if any)	
Conda Init: (Provide path to conda initialization scripts)	
Conda Yaml: (Upload a yaml file to build the conda environment at runtime) No file chosen	
Turn on use of mamba for speeding up conda-yml installs:	
Enable use of new caching mechanism that will store and reuse conda-yml created environments using conda-pack:	
Reservation:	
QoS:	
Working directory:	HOME
Type:	JupyterLab



Hands On: Interactive Computing

- Log onto portal
 - Open terminal window/shell
 - Clone or look for the AI Unlocked GitHub code:
 - <https://github.com/access-ci-org/AI-Unlocked-Workshop-2025>
- Browse files via portal
- Portal Job Builder
 - Pick example and submit
 - Build a new Job
- Launch Jupyter lab
 - Run examples from AI Unlocked GitHub examples
- Launch Jupyter lab



Using HPC-Systems: Summary of Hands-on Exercises

- Part 1:
 - Hands On: Logging onto Expanse: login.expanse.sdsc.edu (S20)
 - Hands-on: Cloning the Repo (S21)
- Part 2:
 - Hands On: Discovering Available Software and Tools (S30)
- Part 3:
 - Hands-on: Simple Batch Script Job: ENV_Info (S40)
 - Hands On: Logging on and checking account info (S47)
 - Hands On: Submitting Simple (ENV) CPU Job to the SLURM Batch Queue (S48)
 - Hands On: Submitting CPU Calc Prime to the Batch Queue (S49)
 - Hands On: Submitting GPU Hello MPI Job (S52)



Thank You!

Q&A

If you have problems, please contact **consult@sdsc.edu**

<https://github.com/sdsc-complecs/interactive-computing/>



Using HPC-Systems: Summary of Hands-on Exercises

- Part 1:
 - Logging onto Expanse:
 - login.expanse.sdsc.edu (S20)
 - Cloning & updating the repo (S21)
- Part 2:
 - Discovering Available Software and Tools (S30)
- Part 3:
 - Simple Batch Script Job: ENV_Info (S40)
 - Logging on and checking account info (S47)
 - Submitting Simple (ENV) CPU Job to the SLURM Batch Queue (S48)
 - Submitting CPU Calc Prime to the Batch Queue (S49)
 - Submitting GPU Hello MPI Job (S52)
- Part 4:
 - Log onto portal
 - Open terminal window/shell
 - Clone or look for the AI Unlocked GitHub code:
 - <https://github.com/access-ci-org/AI-Unlocked-Workshop-2025>
 - Browse files via portal
 - Portal Job Builder
 - Pick example and submit
 - Build a new Job
 - Launch Jupyter lab
 - Run examples from AI Unlocked GitHub examples
 - Launch Jupyter lab



Hands On: Interactive Computing

- Part 4:
 - Log onto portal
 - Open terminal window/shell
 - Clone or look for the AI Unlocked GitHub code:
 - <https://github.com/access-ci-org/AI-Unlocked-Workshop-2025>
 - Browse files via portal
 - Portal Job Builder
 - Pick example and submit
 - Build a new Job
 - Launch Jupyter lab
 - Run examples from AI Unlocked GitHub examples
 - Launch Jupyter lab



SDSC 2025 TRAINING EVENTS

CyberInfrastructure-Enabled Machine Learning - CIML Summer Institute

June 24 – 27, 2025 (Application deadline: Friday, April 11, 2025)

<https://www.sdsc.edu/events/202506-CIML-SI.html>

- A three-day program teaching machine learning (ML) concepts.

HPC and Data Science Summer Institute

August 4 – 7, 2025 (Application deadline: Friday, April 25, 2025)

- A comprehensive week-long workshop that covers introductory-to-intermediate topics in HPC, data science, and artificial intelligence (AI).

Getting Started with HPC

April 30 – May 1 (Registration closes: April 16, 2025)

<https://www.sdsc.edu/events/202504-Getting%20Started%20HPC.html>

- Virtual workshop designed to strengthen skills using Advanced Cyberinfrastructure and shared HPC (High-Performance Computing) systems through presentations, discussions, and hands-on experiences.



Resources

- AI Unlocked GitHub Repo, including this presentation:
 - <https://github.com/sdsc-complecs/interactive-computing/>
- SDSC Training Resources
 - HPC/CI On-Demand Training Catalog: <https://www.sdsc.edu/education/on-demand-learning/index.html>
 - HPC Example Code: <https://github.com/sdsc-hpc-training-org/hpctr-examples>
 - Hands-on HPC/CI Training: <https://hpc-training.sdsc.edu/>
 - Running notebooks
 - Using galyleo: <https://github.com/sdsc/galyleo>
 - Expanse Notebooks Collection: <https://github.com/sdsc-hpc-training-org/Expanse-Notebooks>
- Expanse :
 - Project page: expanse.sdsc.edu
 - User Guide: https://expanse.sdsc.edu/support/user_guides/expanse.html
- Problems? Contact consult@sdsc.edu