Computing on Composable Resources: NSF ACES and NSF FASTER

February 13, 2024

High Performance
Research Computing
DIVISION OF RESEARCH

Outline

- What is composability?
- FASTER and ACES overview
  - Composability
  - Avilalable accelerators
- Documentation and Training
- Getting Started on the ACES and FASTER Clusters
  - Usage Policies and Access
  - Using Accelerators
  - Portal and Cluster Basics
- Software Infrastructure

What is Composability?

Composable HPC Architectures for AI

Common HPC

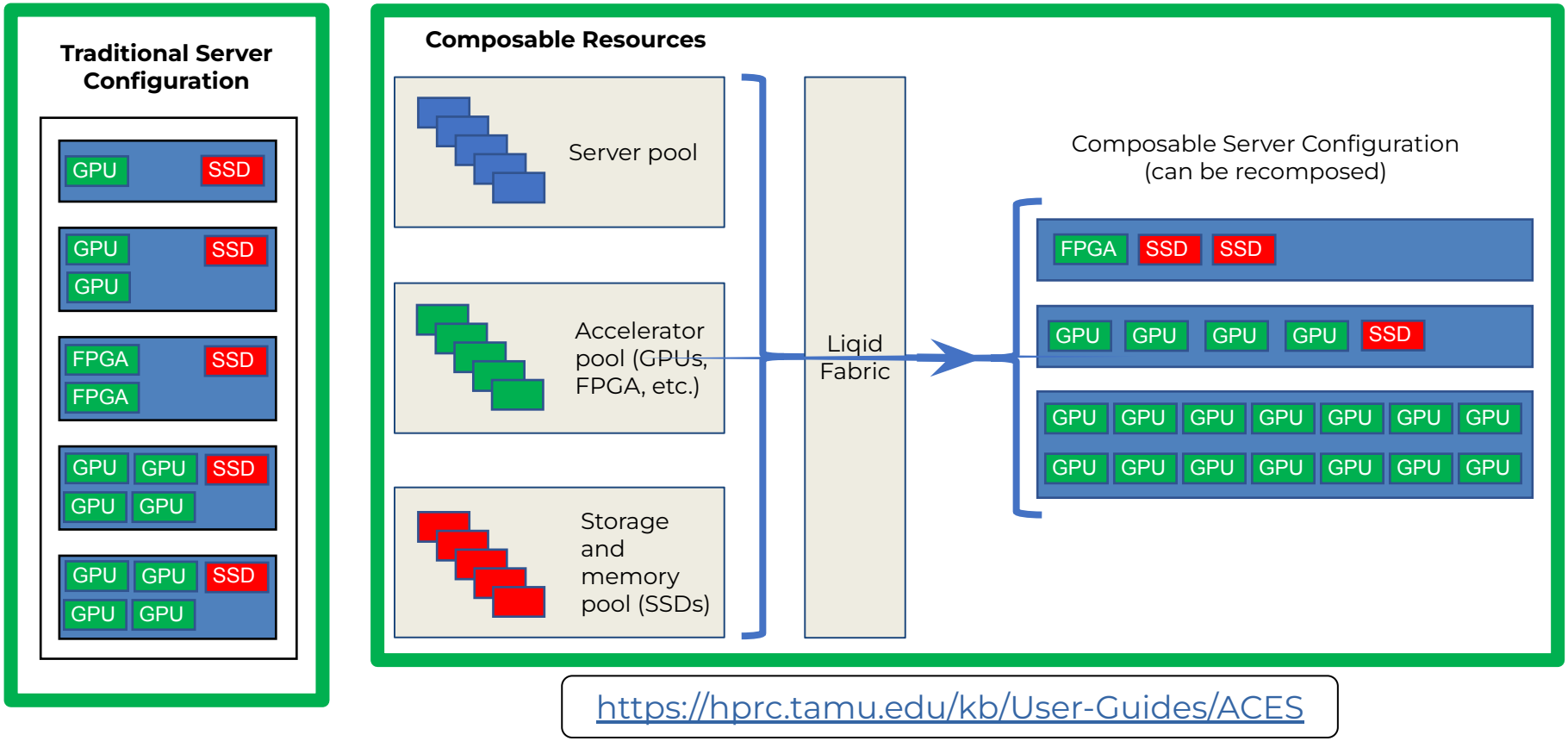
- Built on Converged Hardware
- Static Hardware Design
- Fixed GPU/Accelerator
- Fixed Memory
- Storage: SATA and SAS
- Vendor Lock

HPC for AI

- Built on Disaggregated Hardware
- Composable Hardware Platform
- Composable GPU/Accelerator
- Composable Memory - Optane
- Modern Storage: NVMe-oF
- Open Platform

Next Generation HPC/AI Platform Supports Composable Accelerators and Memory

Composability - 2023



https://hprc.tamu.edu/kb/User-Guides/ACES

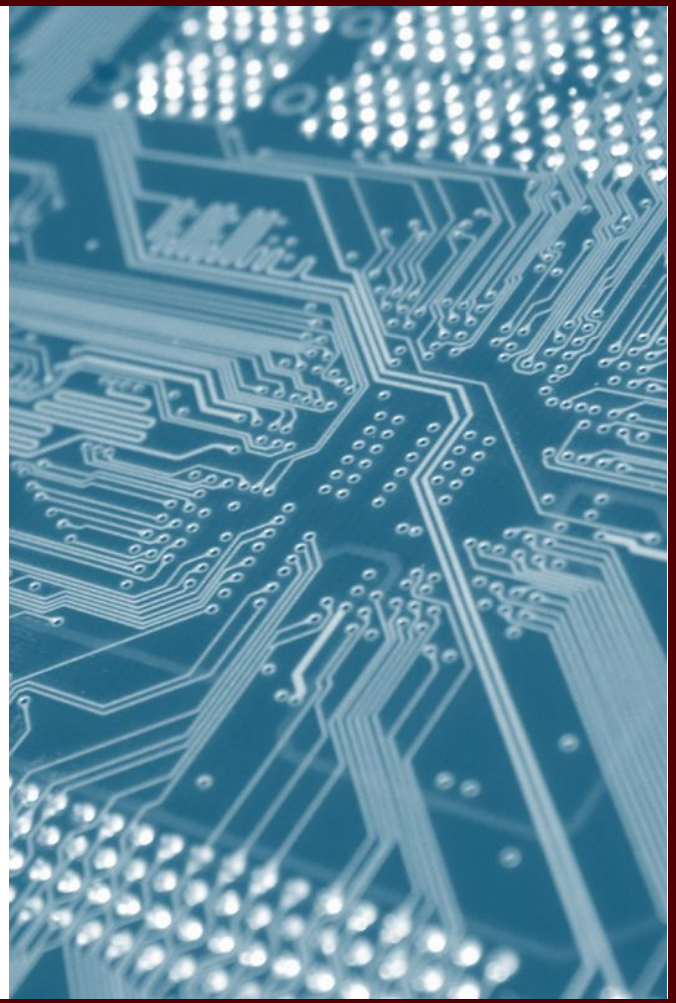
FASTER and ACES!

HPRC's Composable Clusters

- **FASTER** – First large-scale composable CPU/GPU system
- **ACES** – Composability for mixed-resource workflows

Note:

- contact help@hprc.tamu.edu to request a given composable configuration for a cluster.
- In-progress: making Slurm and Liqid handle it automatically

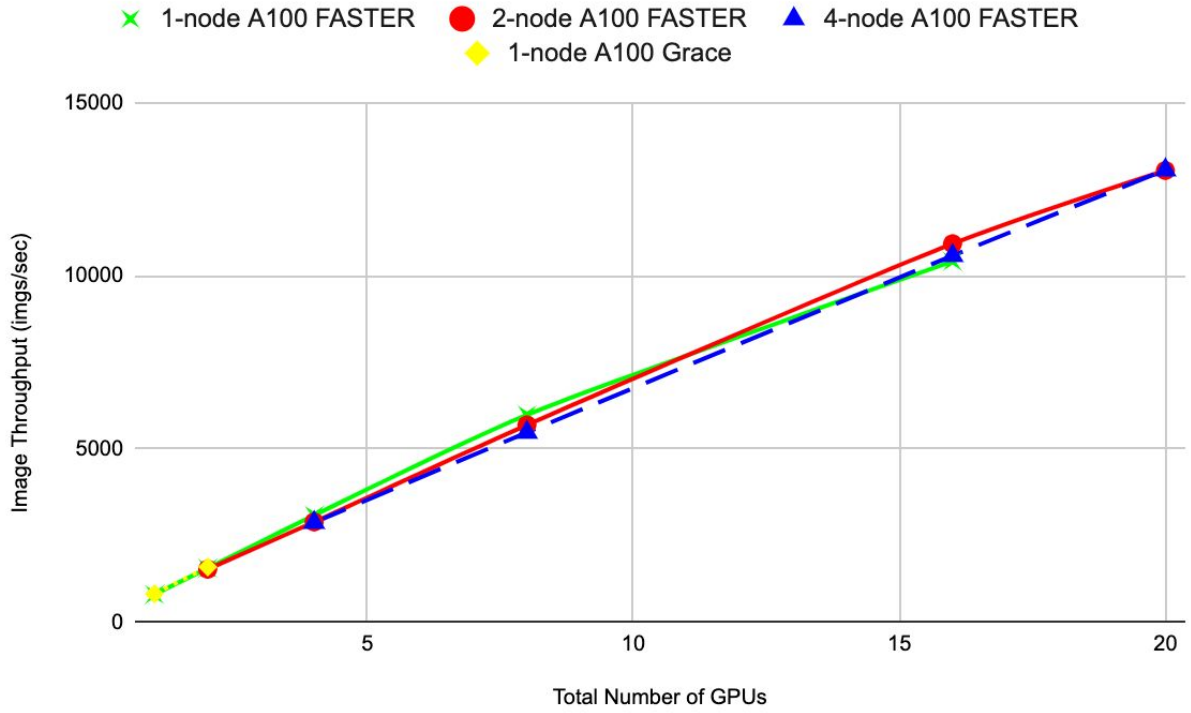


FASTER (Fostering Accelerated Scientific Transformations, Education, and Research)

https://hprc.tamu.edu/kb/User-Guides/FASTER

Node Type	Quantity
64-core login nodes	3
64-core compute nodes (256GB RAM each)	180 (11,520 cores)
Composable GPUs	200 T4 16GB, 40 A100 40GB 8 A10 24GB, 4 A30 24GB and 8 A40 48GB
Interconnect	Mellanox HDR100 InfiniBand (Liqid PCIe Gen4 (GPU composability)
Global Disk	5PB DDN Lustre appliances

Scaling on Composable Fabrics



BERT-Large Language Model Fine-Tuning

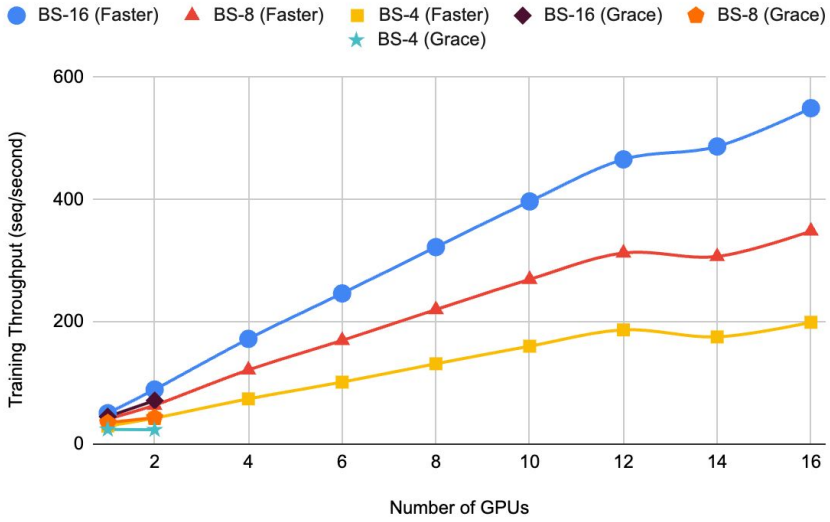


Figure (a) The BERT-Large scaling behavior

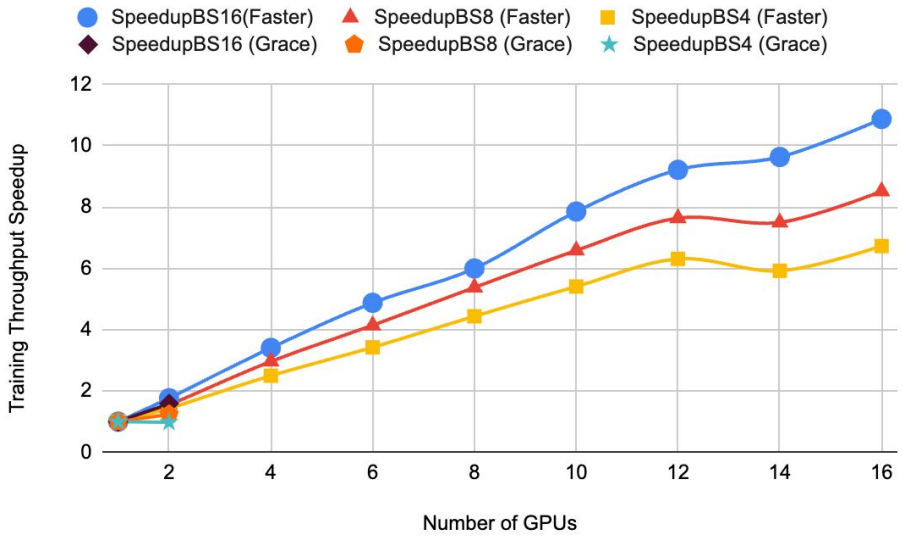
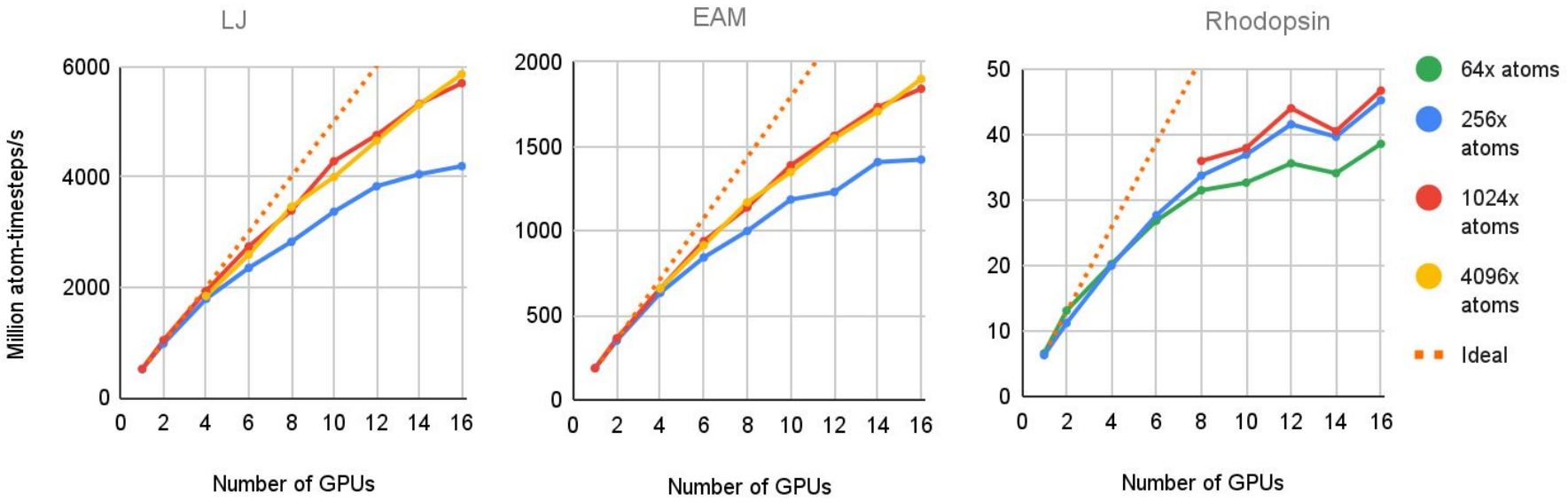


Figure (b) throughout speedup with different batch sizes (BS) as number of GPUs increase.

LAMMPS - 16 A100 GPUs on a Single Node



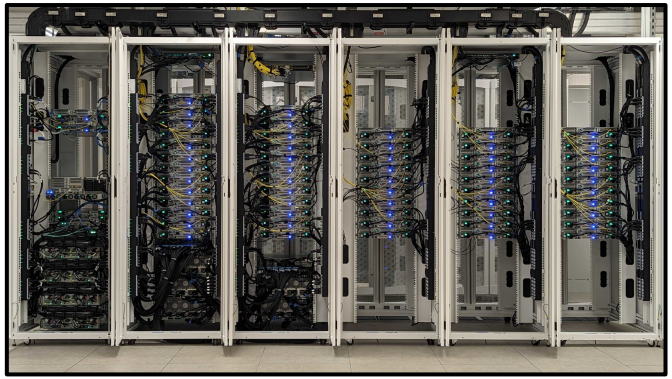
NSF ACES

Accelerating Computing for Emerging Sciences

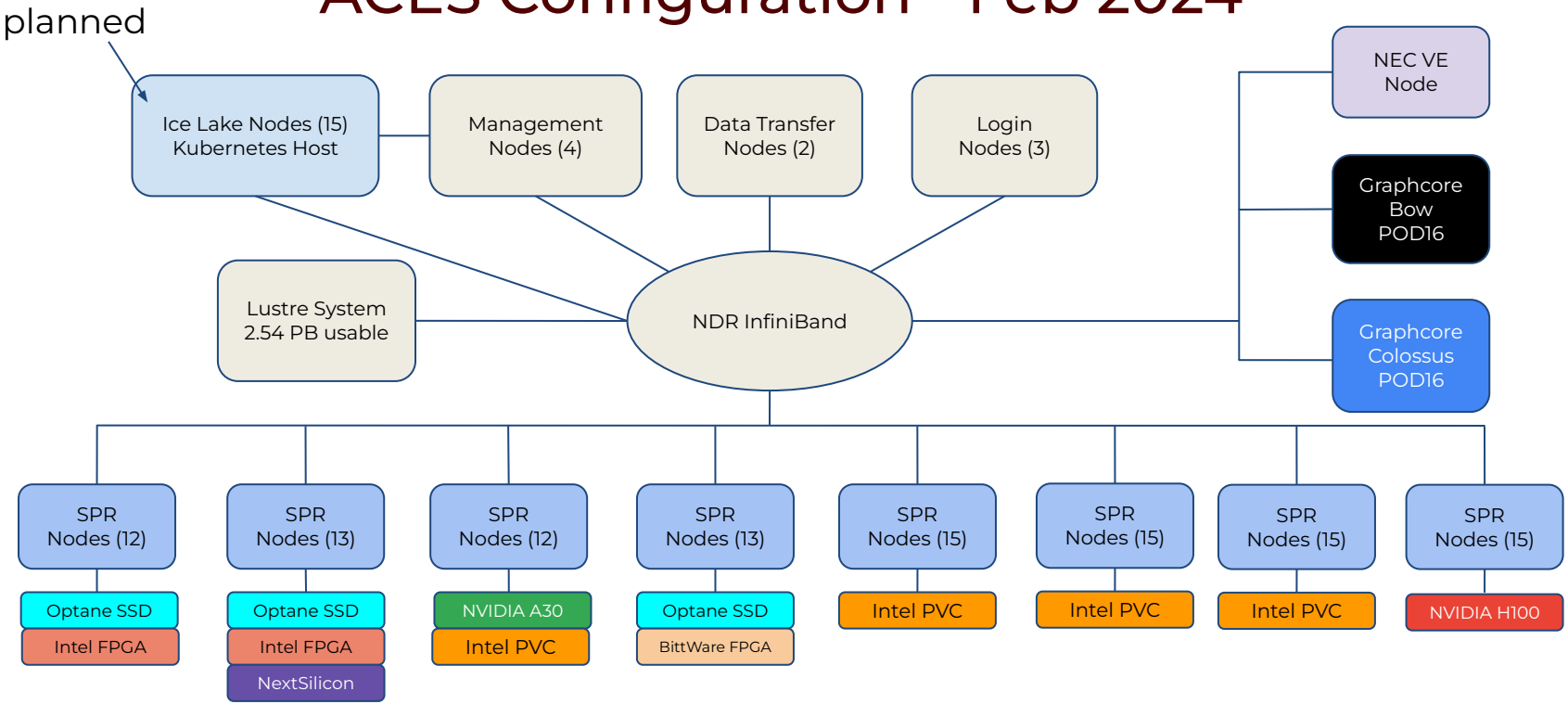
Our Mission:

- NSF ACSS CI test-bed
- Offer an accelerator testbed for numerical simulations and **AI/ML workloads**
- Provide consulting, technical guidance, and training to researchers
- Collaborate on computational and data-enabled research.

ACES In Action



ACES Configuration - Feb 2024



ACES System Description

Component	Description
CPU-centric computing with variable memory requirements	Dual Intel Sapphire Rapids 2.1 GHz **96** cores per node, 512 GB memory, 1.6 TB NVMe storage (PCIe 5.0), NVIDIA Mellanox NDR 200 Gbps InfiniBand
Composable infrastructure	Reconfigurable infrastructure that allows up to 20 PCIe cards (GPU, FPGA, VE, etc.) per compute node
Data transfer nodes	100 Gbps network adapter

ACES Accelerators

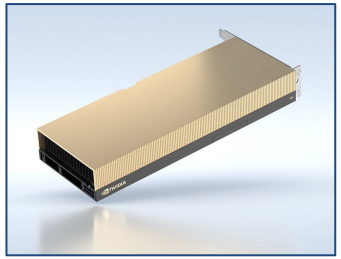
Component	Quantity	Description
Graphcore IPU	32	16 Colossus GC200 IPUs; 16 Bow IPUs. Each IPU group hosted with a CPU server as a POD16 on a 100 GbE RoCE fabric
Intel PAC D5005 FPGA	2	Accelerator with Intel Stratix 10 GX FPGA and 32 GB DDR4
BittWare IA-840F FPGA	2	Accelerator with Agilex AGF027 FPGA and 64 GB of DDR4
NextSilicon Coprocessor	2	Reconfigurable accelerator with an optimizer continuously evaluating application behavior.
NEC Vector Engine	8	Vector computing card (8 cores and HBM2 memory)
Intel Optane SSD	48	18 TB of Intel Optane SSDs addressable as memory w/ MemVerge Memory Machine.
NVIDIA H100 + A30	30 + 4	NVIDIA GPUs for HPC, DL Training, AI Inference
Intel GPU Max 1100 (PVC)	120	Intel GPUs for HPC, DL Training, AI Inference

Research Workflows - Accelerators

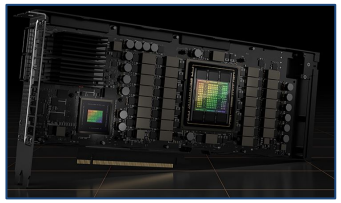
Hardware Profile	Applications Supported	
NEC Vector Engines	• AI/ML (Statistical Machine Learning, Data Frame) • Chemistry (VASP, Quantum ESPRESSO) • Earth Sciences • NumPy Acceleration	• Oil & Gas (Seismic Imaging, Reservoir Simulation) • Plasma Simulation • Weather/Climate Simulation
Graphcore IPUs	• Graph Data • LSTM Neural Networks	• Markov Chain Monte Carlo • Natural Language Processing (Deep Learning)
Intel/Bittware FPGA	• AI Models for Embedded Use Cases • Big Data • CXL Memory Interface • Deep Learning Inference • Genomics	• MD Codes • Microcontroller Emulation for Autonomy Simulations • Streaming Data Analysis
Intel Optane SSDs	• Bioinformatics • Computational Fluid Dynamics (OpenFOAM)	• MD Codes • R • WRF
NextSilicon	• Biosciences (BLAST) • Computational Fluid Dynamics (OpenFOAM) • Cosmology (HACC) • Graph Search (Pathfinder)	• Molecular Dynamics (NAMD, AMBER, LAMMPS) • Quantum ChromoDynamics (MILC) • Weather/Environment modeling (WRF)

More About the Accelerators

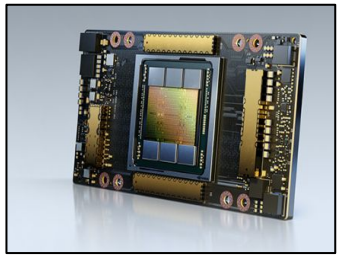
GPUS



A30s: Support less intense workloads relying on numerical simulations and AI/ML methods.



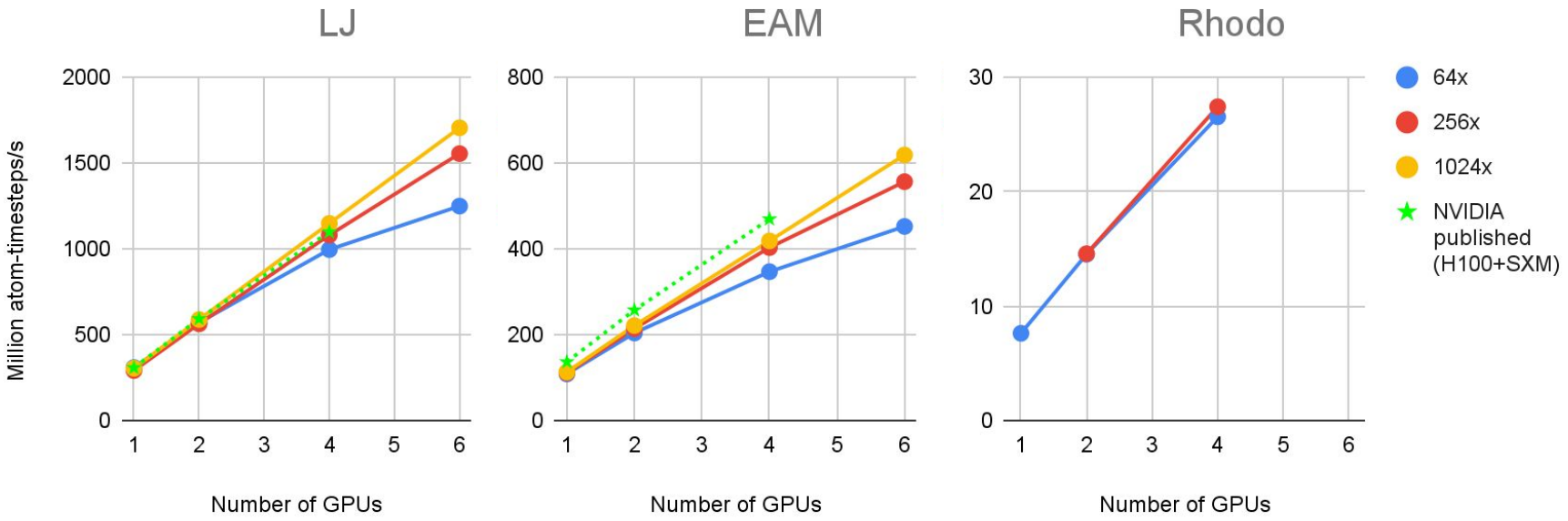
H100s: Supports computationally intensive workloads employing numerical simulations and AI/ML methods. ACES-only.



A100s: Somewhere in-between!
FASTER only.

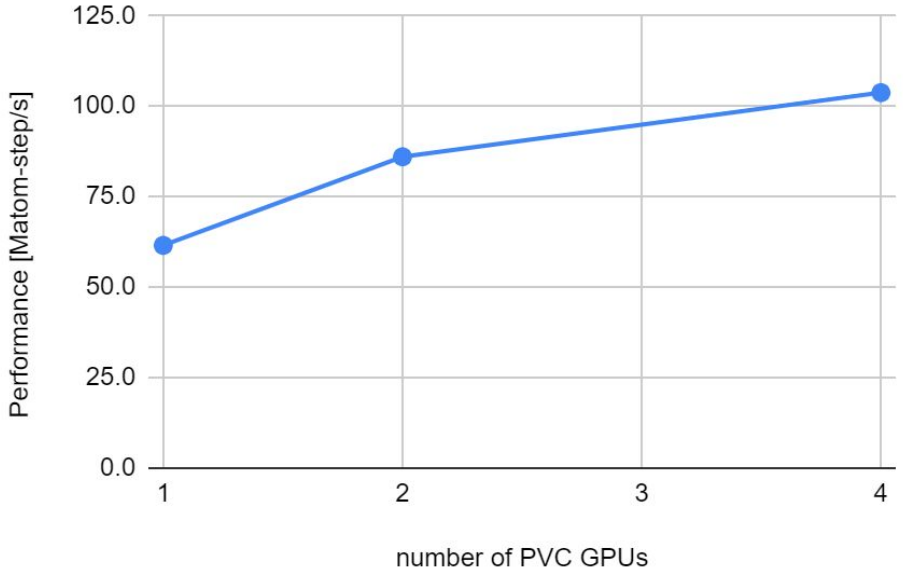
GPUs: LAMMPS on Composed A30s

NVIDIA container: `nvcr.io/hpc/lammps:patch_15Jun2023`



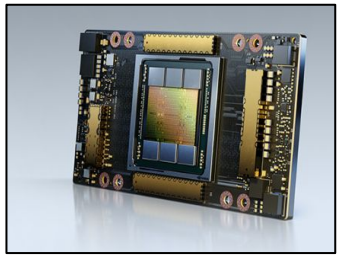
GPUs: LAMMPS on Intel PVCs

- Liquid Crystal
- Gay-Berne benchmark
- biaxial ellipsoid mesogens in isotropic phase
- GPU package, 16 cores (fixed)
- LAMMPS (3 Aug 2023 - Development branch)

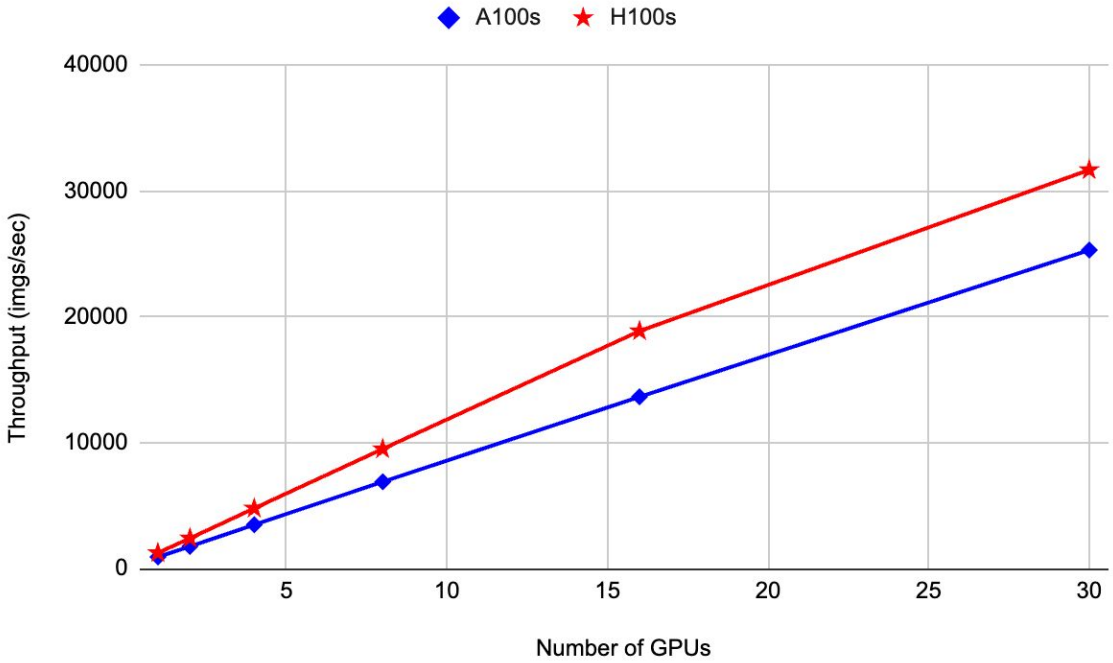


GPUs: ResNet50 on H100 vs A100

A100

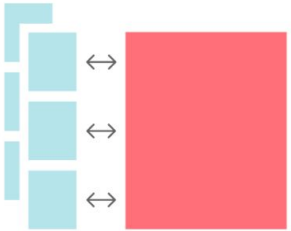
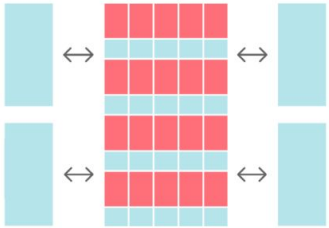
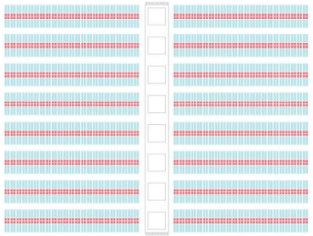


H100



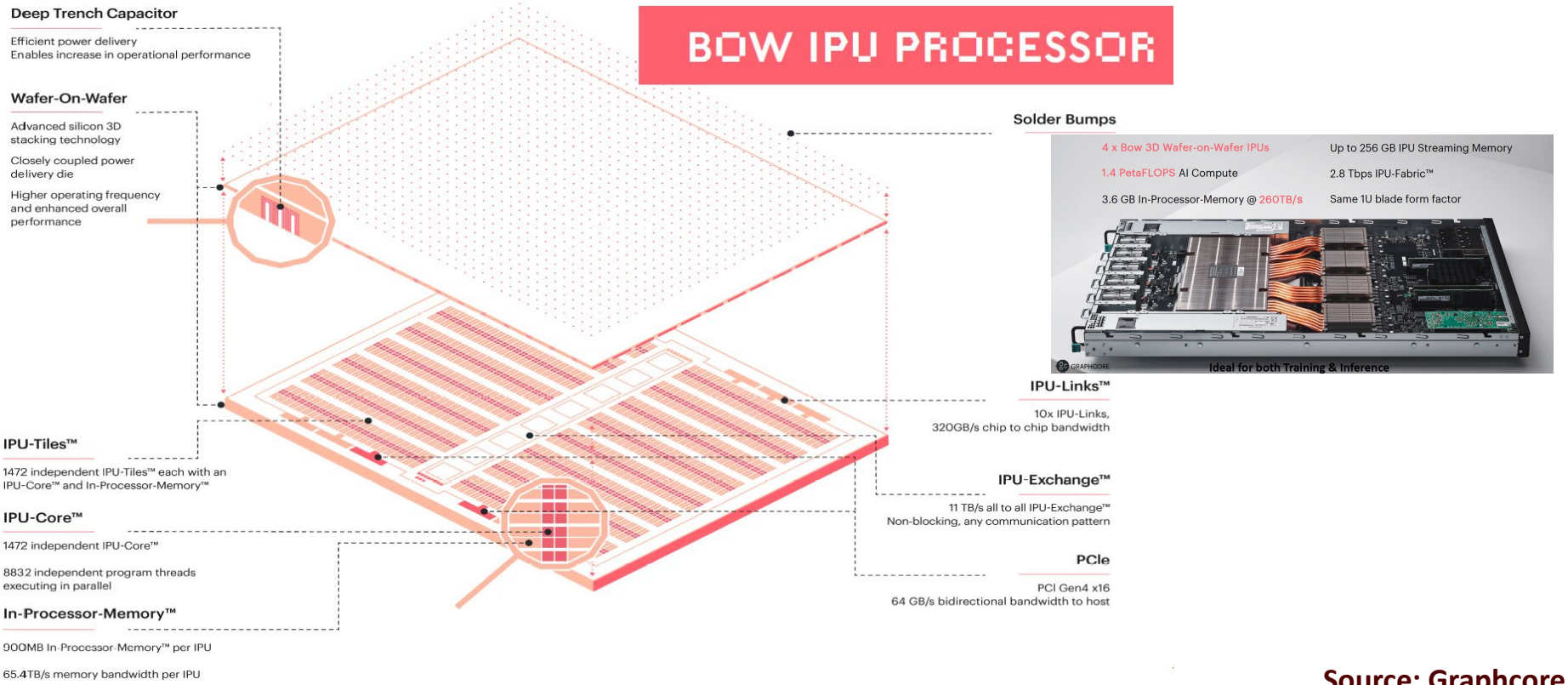
- Horovod TensorFlow ResNet 50

Graphcore IPUs

	CPU	GPU	IPU
Parallelism	Designed for scalar processes	SIMD/SIMT architecture. Designed for large blocks of dense contiguous data	Massively parallel MIMD. Designed for fine-grained, high-performance computing
			
Memory Access	Off-chip memory	Model and data spread across off-chip and small on-chip cache, and shared memory	Model and data tightly coupled, and large locally distributed SRAM

www.graphcore.ai/bow-processors

Graphcore IPUs

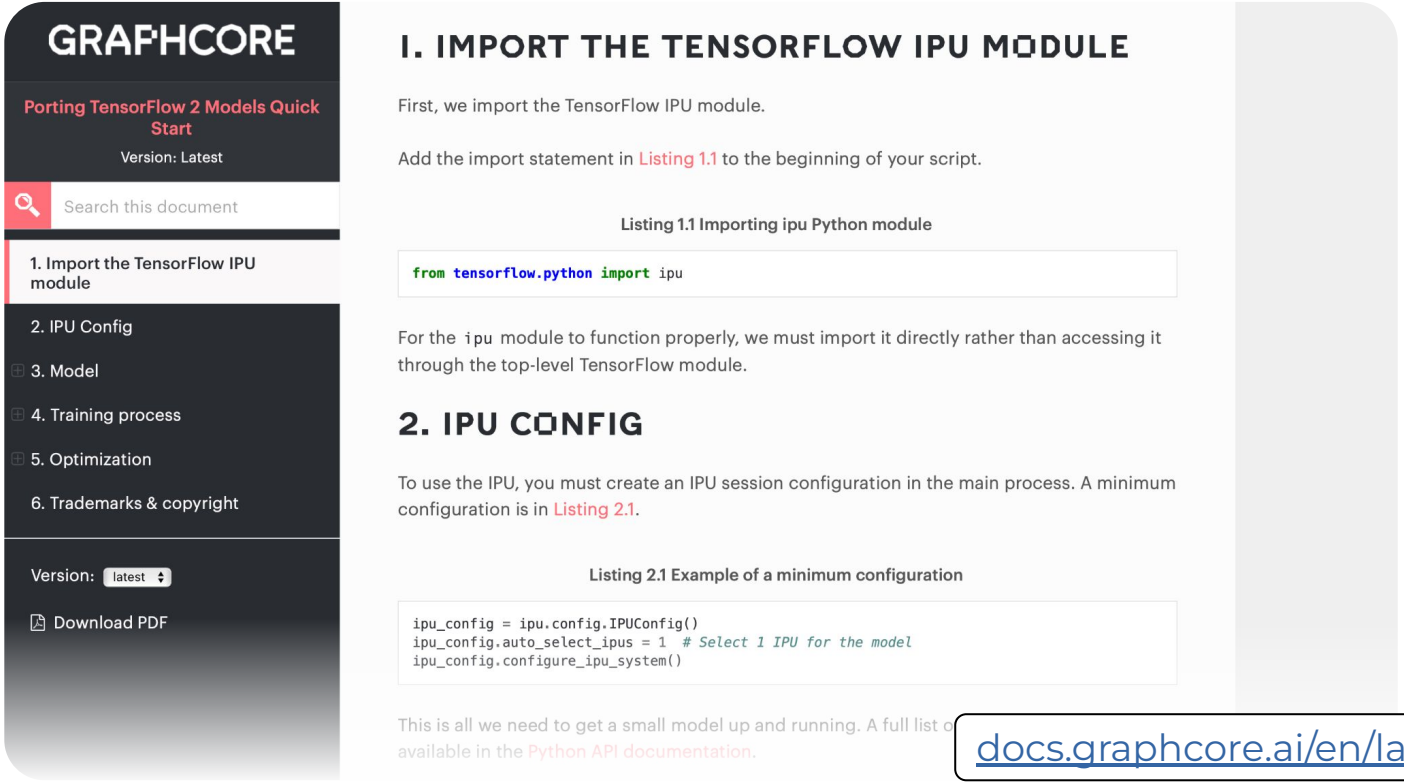


Source: Graphcore

Graphcore IPU Systems

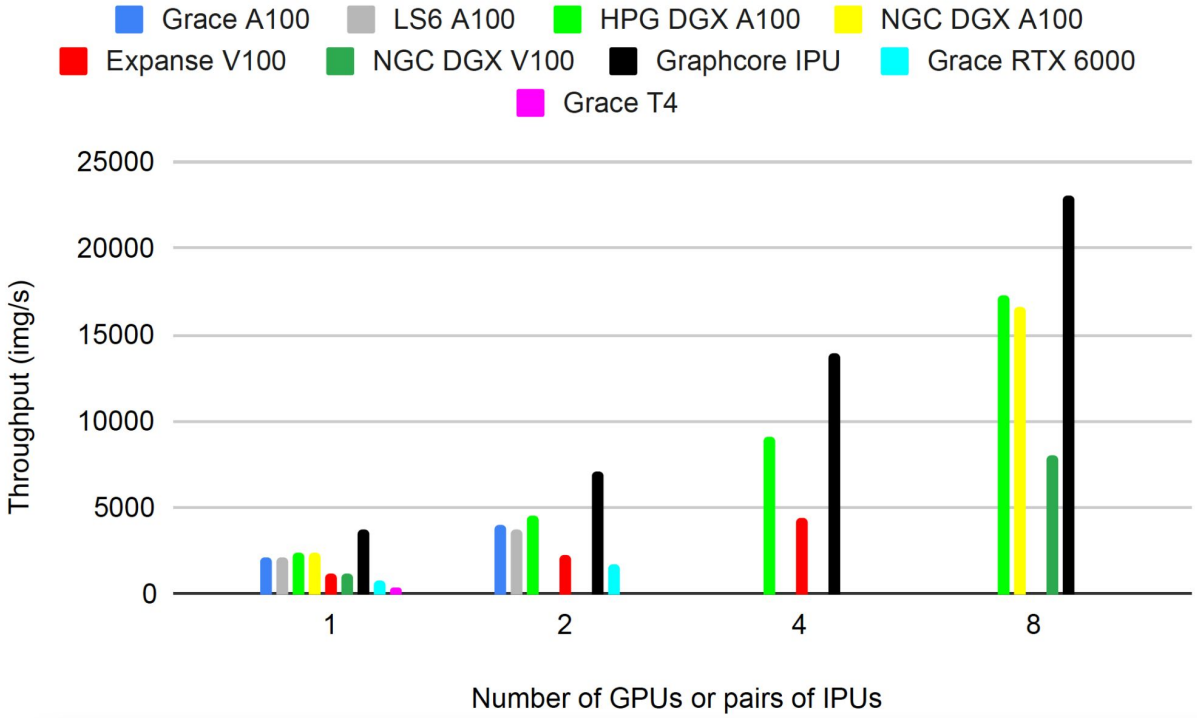
POD Type	Machine Type	IPU Frequency	Aggregate Performance
Bow-2000 POD16	4 Bow-2000	1.85 GHz	5.6 PFLOPS FP16.16 1.4 PFLOPS FP32
Colossus POD16	4 IPU-M2000	1.325 GHz	4 PFLOPS FP16.16 1 PFLOPS FP32

Graphcore IPUs



docs.graphcore.ai/en/latest

GraphCore IPUs vs GPUs: PyTorch ResNet50

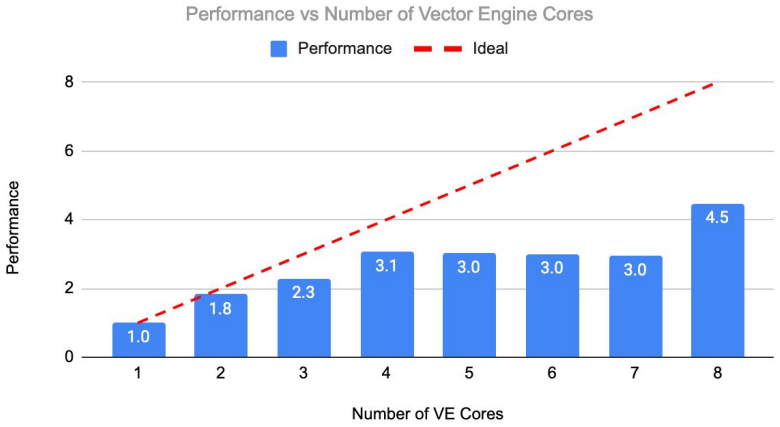


Abhinand S. Nasari, et al.. 2022. Benchmarking the Performance of Accelerators on National Cyberinfrastructure Resources for Artificial Intelligence / Machine Learning Workloads. In Practice and Experience in Advanced Research Computing (PEARC '22), July 10-14, 2022, Boston, MA, USA. ACM, New York, NY, USA, 13 Pages. https://doi.org/10.1145/3491418.3530772

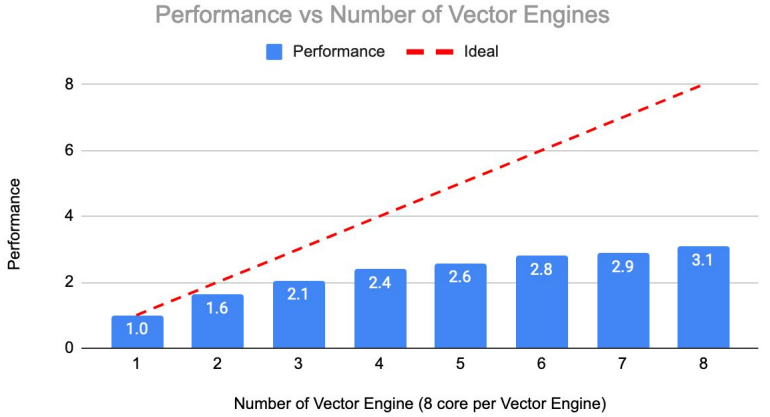
NEC Vector Engine

- 300W PCIe Gen3 x16 card
- 8 cores, 48 GB HBM2 with 1.5 TB/s memory bandwidth
- 2.45 TFLOPS FP64 peak performance
- Eight (8) NEC Vector Engine Type 20B-P cards hosted in a Dell DSS8440 server
- Early testing performed with VASP

NEC Vector Engine



Single Vector Engine

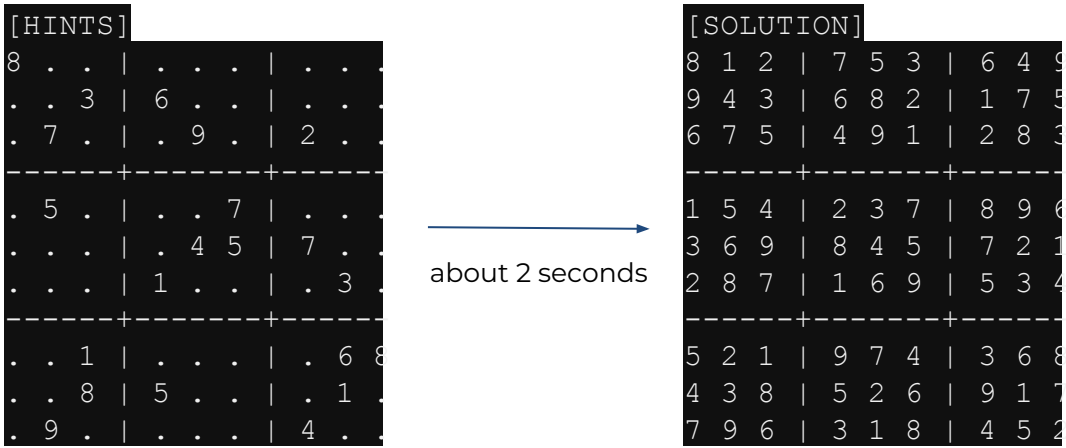


Across Multiple Vector Engines

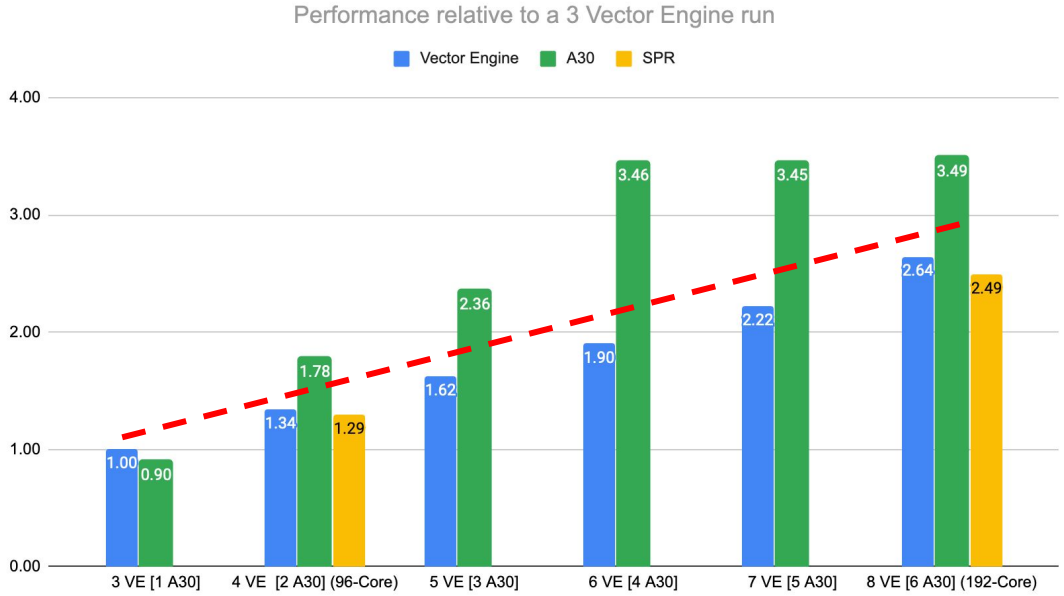
- Plane wave calculations using VASP 6.3.2
- 12 atom supercell for hafnium oxide (HfO_2)

NEC Vector Annealing

- NEC **Vector Annealing** software solves binary optimization problems
- Vector Annealing can be used to simulate Quantum Annealing
  - Quantum Annealing is how a D-Wave quantum computer works
- or you can just play sudoku



VASP on GPUs, CPUs, and Vector Engines



- VASP 6.3.2 for HfO_2, a 96 atom system
- Each Vector Engine has 8 Vector Cores, NVIDIA A30 GPUs composed to a single node, and a Sapphire Rapid nodes have 96 cores

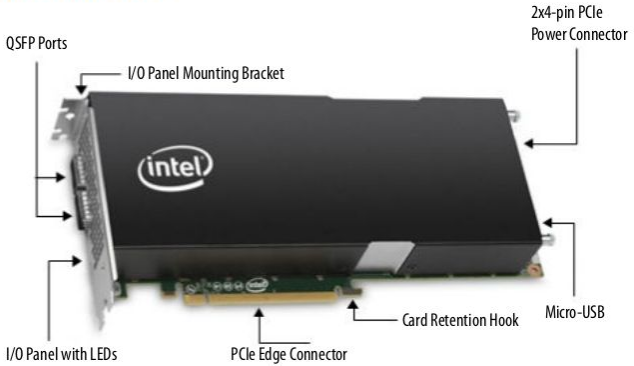
NextSilicon Coprocessors

- 250W PCIe Gen5 x8 card
- 64 GB HBM2e memory with 1.6 TB/s memory bandwidth
- Maverick Generation 1 coprocessor reconfigurable with an optimizer continuously evaluating application behavior
- Hardware integer compute units
- 2 TFLOPS FP64 peak performance using software emulated floating point operations.


```
% check_pci_buses.sh ac018
=====================================
ac018
=====================================
0000:22:00.0-[23-38]--+-00.0-[24-37]----00.0-[25-37]----10.0-[26-37]----00.0-[27-37]--+-00.0-[28]----00.0  NextSilicon Ltd Maverick
                      |                                                               +-01.0-[29]--
                      |                                                               +-02.0-[2a]--
                      |                                                               +-03.0-[2b]--
                      |                                                               +-04.0-[2c]--
                      |                                                               +-05.0-[2d]--
                      |                                                               +-06.0-[2e]--
                      |                                                               +-07.0-[2f]--
                      |                                                               +-08.0-[30]--
                      |                                                               +-09.0-[31]--
                      |                                                               +-0a.0-[32]--
                      |                                                               +-0b.0-[33]--
                      |                                                               +-0c.0-[34]--
                      |                                                               +-0d.0-[35]--
                      |                                                               +-0e.0-[36]--
                      |                                                               \-0f.0-[37]--
                      \-1c.0-[38]----00.0  Broadcom / LSI PEX880xx PCIe Gen 4 Switch
```

FPGAs

Intel FPGA PAC D5005



BittWare IA-840F FPGA

- Supported Tools:
  - Intel FPGA OneAPI
  - Intel Quartus Prime

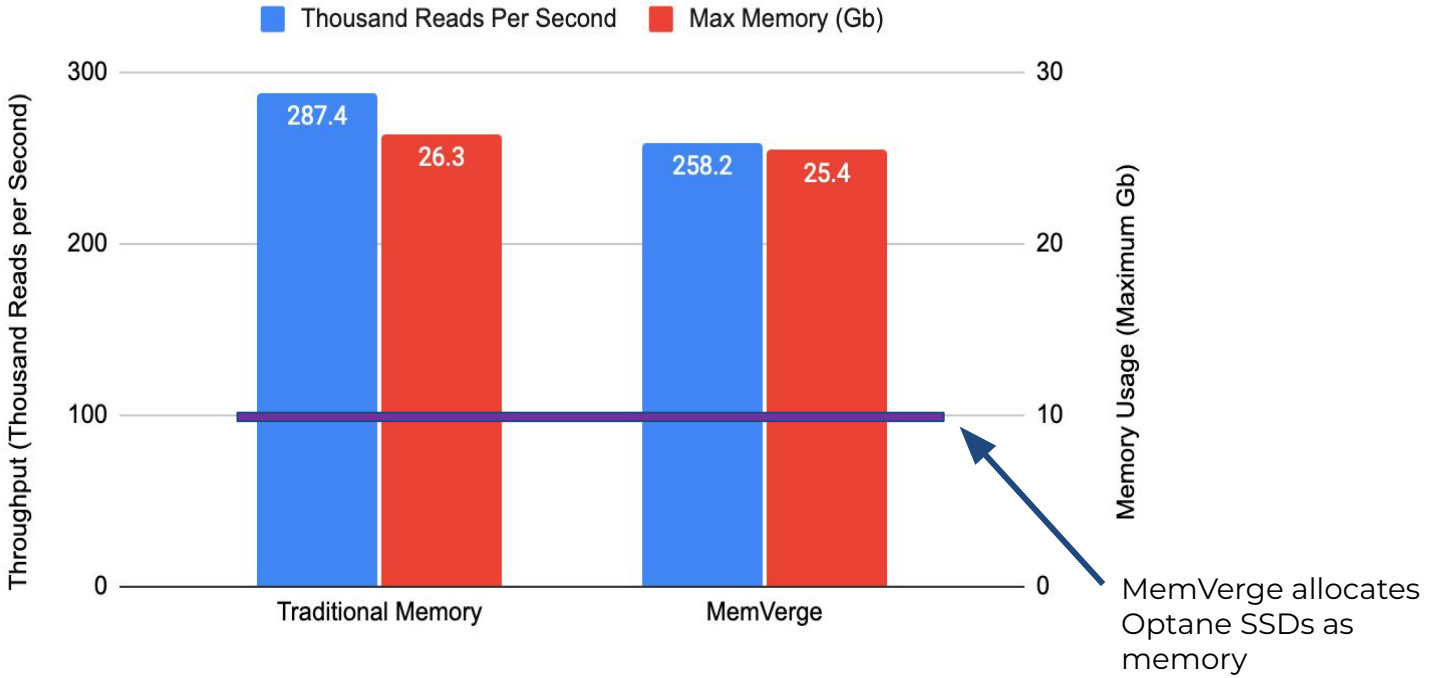
Intel D5005 FPGA

- Supported Tools:
  - Intel Acceleration Stack
  - Intel Quartus Prime
  - Open Programmable Acceleration Engine

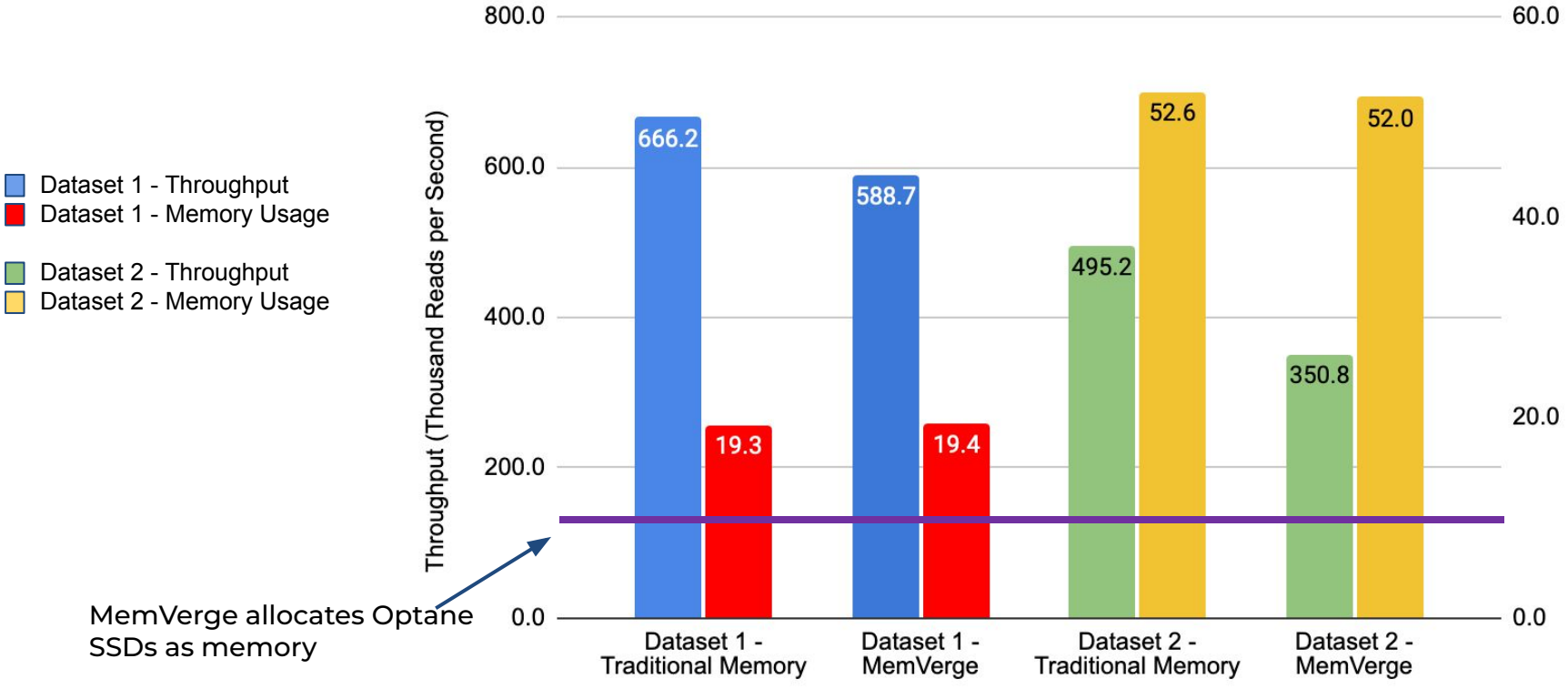
Composing Memory on Nodes

- Intel Optane SSDs with MemVerge
- Tadpole (from BBMap) – a memory-intensive pipeline for DNA sequencing read correction
- Memory utilization scales with data size for 2 data sets
- Configured to allocate Optane SSD as memory when memory usage > 10 Gb

Composable Memory – DNA Sequencing and Error Correction with Tadpole



Composable Memory – DNA Sequencing and Error Correction with Tadpole

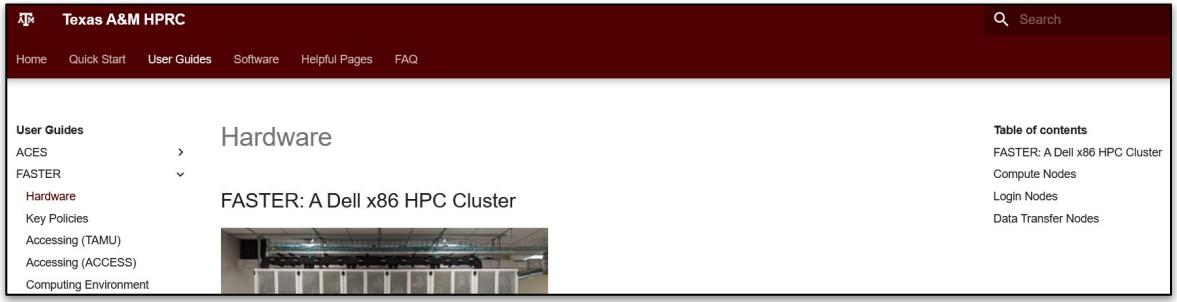


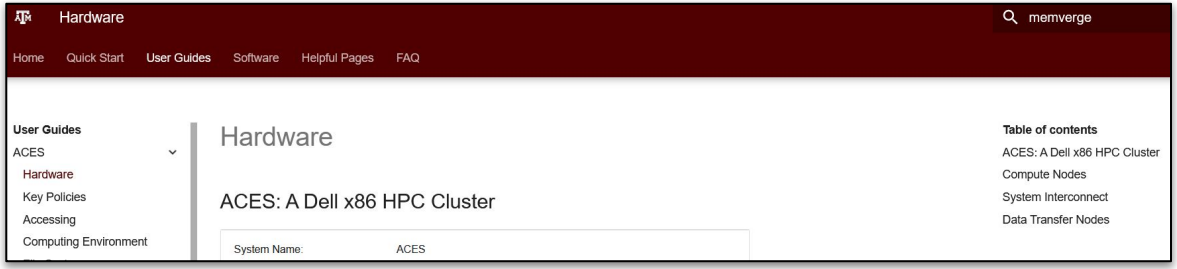
Getting Trained

HPRC Knowledge Base

Knowledge Base for announcements, more hardware details, and more about the subjects we cover today.

hprc.tamu.edu/kb
hprc.tamu.edu/kb/User-Guides/FASTER
hprc.tamu.edu/kb/User-Guides/ACES





Training on YouTube

Texas A&M HPRC

@TexasAMHPRC · 1.35K subscribers · 107 videos

The Texas A&M HPRC YouTube channel assists HPR

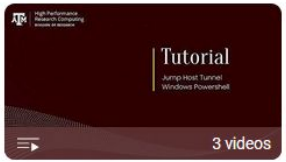
hprc.tamu.edu and 1 more link

Subscribe

https://www.youtube.com/texasamhprc

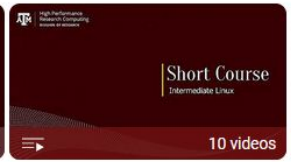
HOME VIDEOS PLAYLISTS COMMUNITY CHANNELS ABOUT

Created playlists



ACES: Getting Started

View full playlist



ACES

View full playlist

Project Account Management (AMS)

View full playlist

Previous Primers and Short Courses:

View full playlist

BRICCs

View full playlist

HPRC OpenOnDemand Portal

View full playlist

Graphcore IPU Training

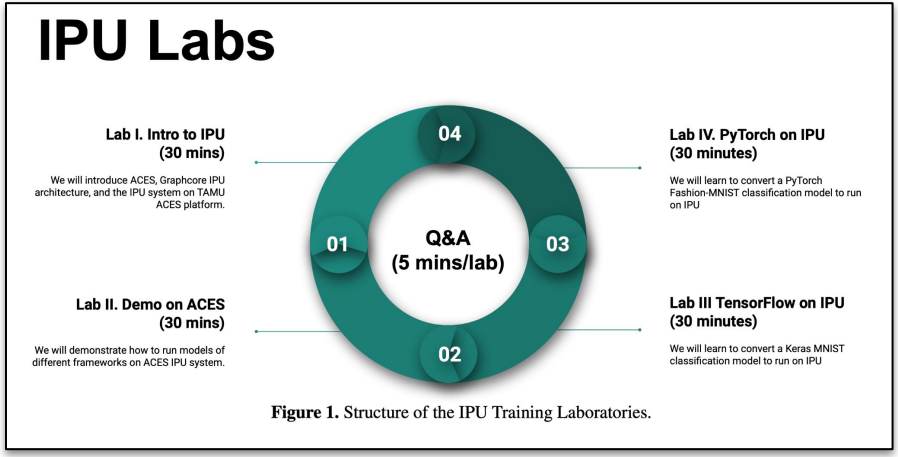
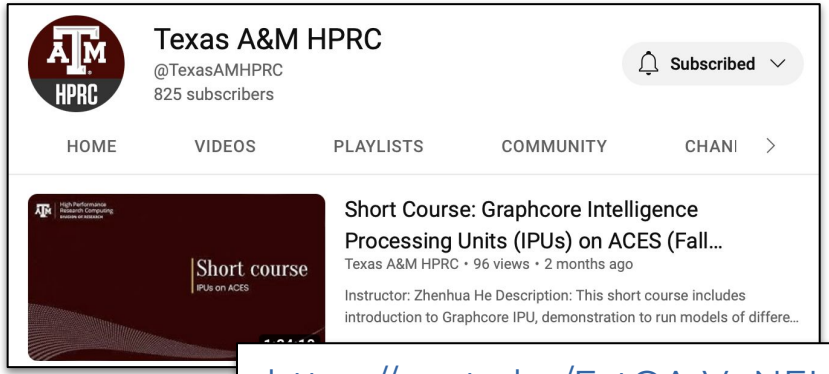


Figure 1. Structure of the IPU Training Laboratories.



https://youtu.be/E_tQA-VuNEU

https://hprc.tamu.edu/training/aces_ipus.html

ACES: GRAPHCORE IPU TUTORIAL

Overview

Instructor:

Time: planned for March 5th

Location: Online using Zoom

Agenda

There are a total of four lab sessions:

1. **Intro to IPU** (30 mins)
   We will introduce Graphcore, IPU architecture, and the IPU system on TAMU ACES platform.

Getting Started on ACES and FASTER

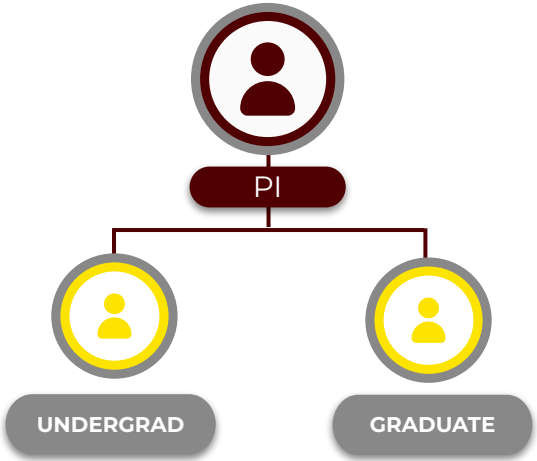
Usage Policies
(Be a good compute citizen)

- It is illegal to share computer passwords and accounts
- Clusters must not be used in any manner that violates the United States export control laws and regulations, EAR & ITAR
- Abide by the license restrictions when using commercial software

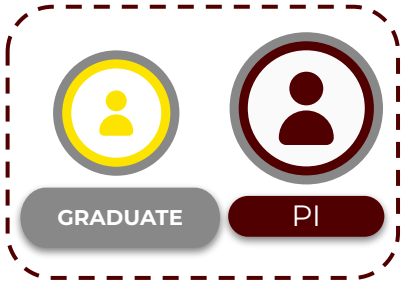
hprc.tamu.edu/policies

Getting on ACES

- You must have an ACCESS account!
- Application for ACES is available through ACCESS: https://allocations.access-ci.org
- Email us at help@hprc.tamu.edu for questions, comments, and concerns.



PIs can apply for an account and sponsor accounts for their researchers.



(Grad students may also apply directly with a letter of collaboration from their PI)

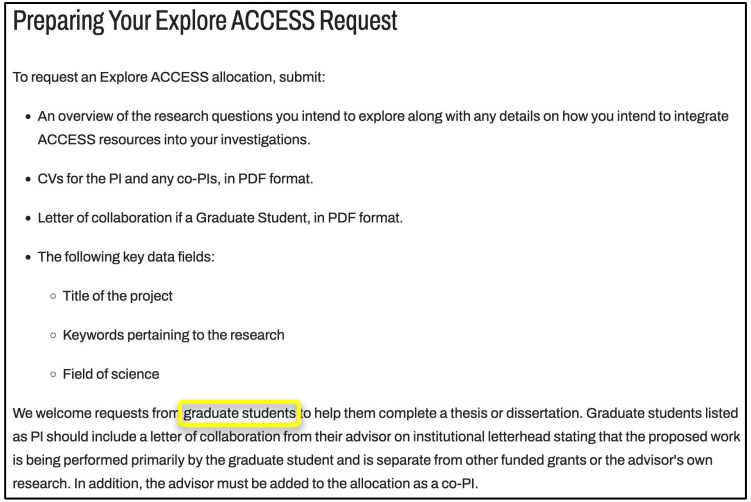
Getting on FASTER

- Researchers can apply for an account on FASTER through ACCESS: https://allocations.access-ci.org

- Texas A&M Researchers can also apply for an account on FASTER via Texas A&M HPRC Application: https://hprc.tamu.edu/apply

ACCESS Allocations Opportunities

See also:
https://hprc.tamu.edu/policies/allocations.html



Preparing Your Explore ACCESS Request

To request an Explore ACCESS allocation, submit:

- An overview of the research questions you intend to explore along with any details on how you intend to integrate ACCESS resources into your investigations.
- CVs for the PI and any co-PIs, in PDF format.
- Letter of collaboration if a Graduate Student, in PDF format.
- The following key data fields:
  - Title of the project
  - Keywords pertaining to the research
  - Field of science

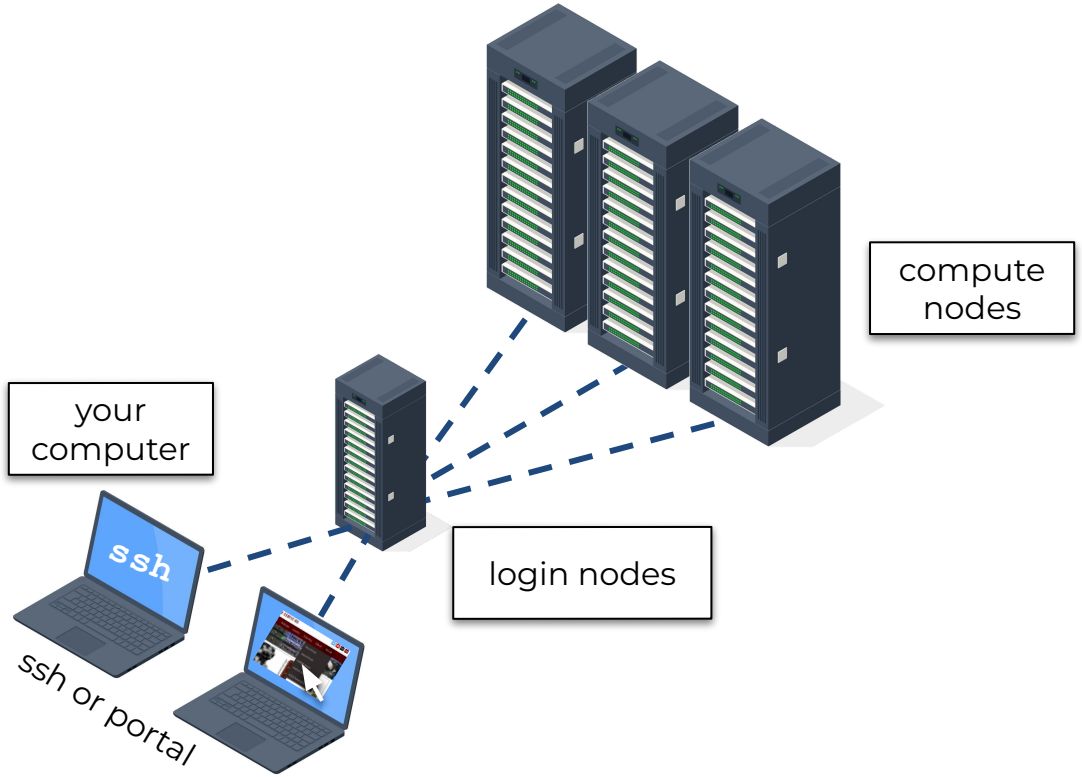
We welcome requests from graduate students to help them complete a thesis or dissertation. Graduate students listed as PI should include a letter of collaboration from their advisor on institutional letterhead stating that the proposed work is being performed primarily by the graduate student and is separate from other funded grants or the advisor's own research. In addition, the advisor must be added to the allocation as a co-PI.

	Explore	Discover	Accelerate
Purpose	Resource Evaluations, grad student projects, small classes, etc.	Large classes, benchmarking at-scale, Campus Champions	Multi-grant programs, Collaborations, Growing gateways
Requests Accepted	Continuously; multiple requests allowed		
Review requirements	Overview	1-page proposal	3-page (max) proposal

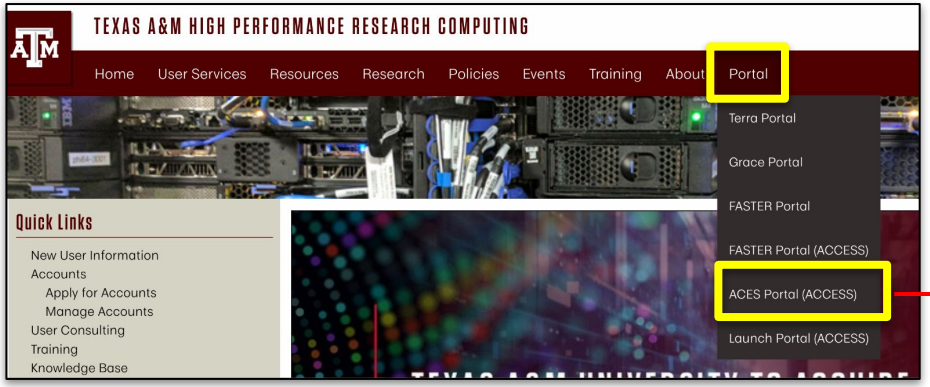
Batch Computing on Clusters

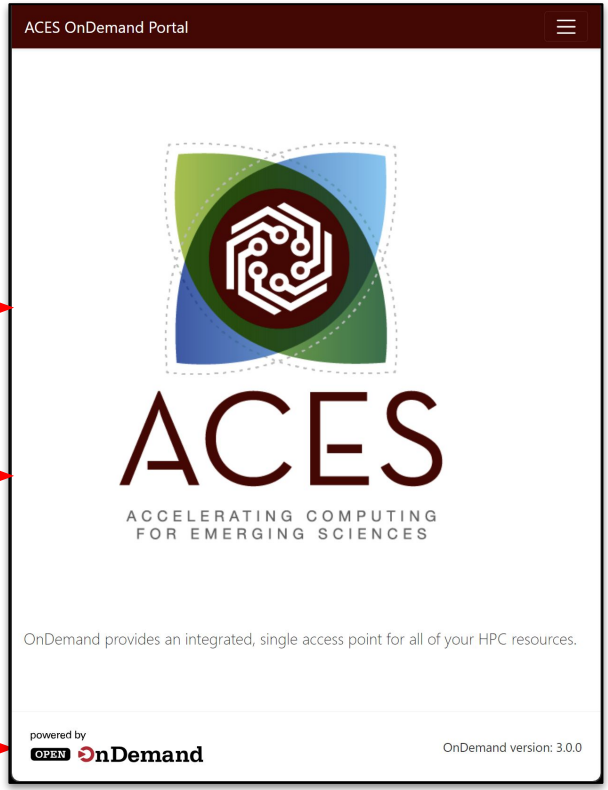
Workflow on a cluster:

- Interact via **your own machine**
- Log in to the cluster's **portal** (and/or the **login nodes**) and write instructions
- Send instructions to **compute nodes** to do the heavy-lifting



ACES Portal





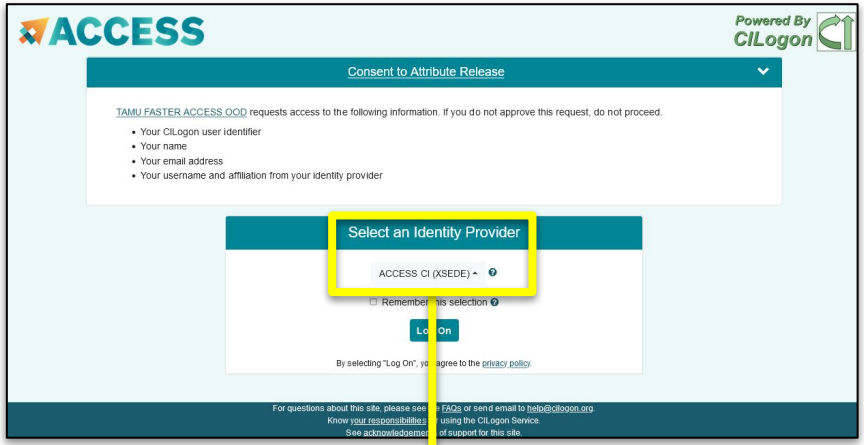
ACES Portal portal-aces.hprc.tamu.edu
is the web-based user interface for the ACES cluster

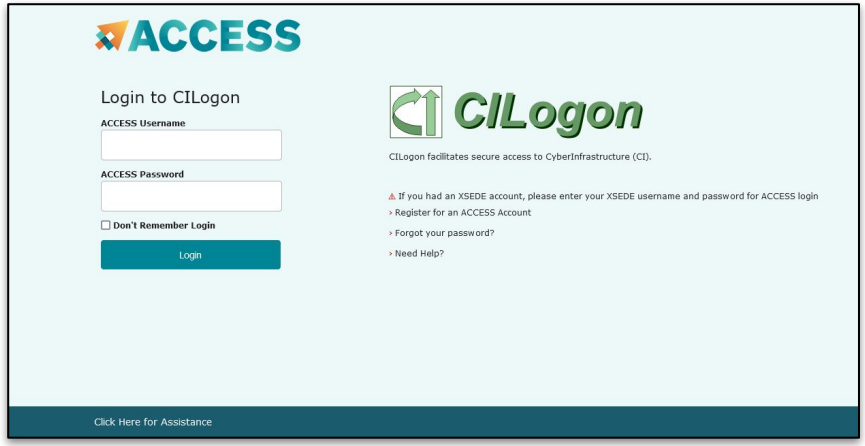
[HPRC Portal YouTube tutorials]

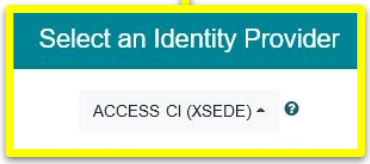
Open OnDemand (OOD) is an advanced web-based graphical interface framework for HPC users

Accessing via ACCESS

Log-in using your ACCESS CI credentials.



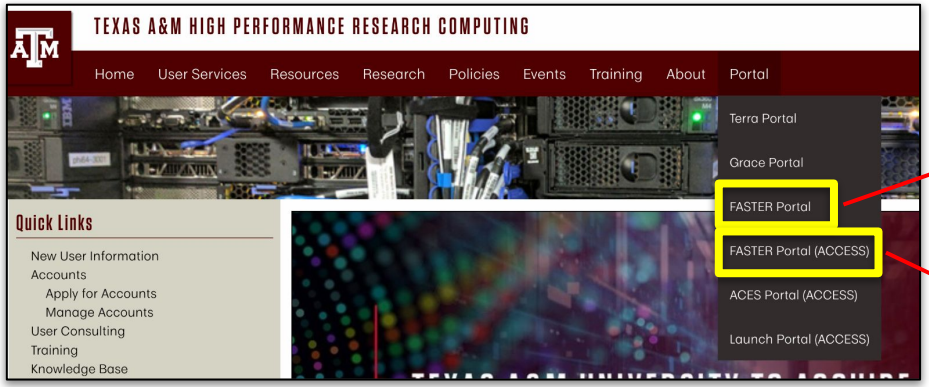


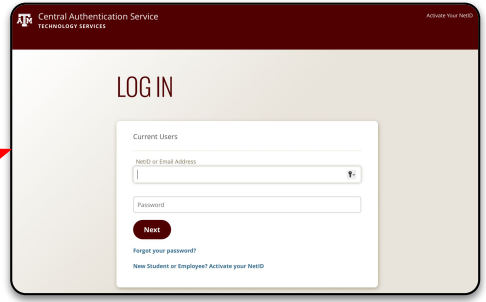


Select the Identity Provider appropriate for your account.

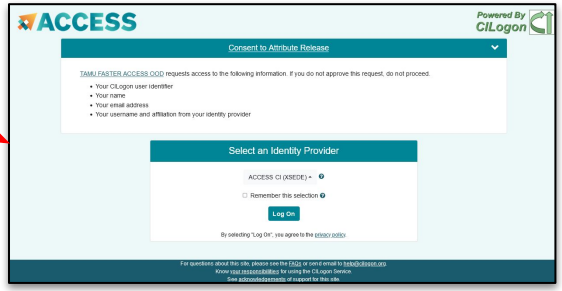
FASTER Portal

TAMU users can log in with their NetID (using VPN if off-campus)





OR



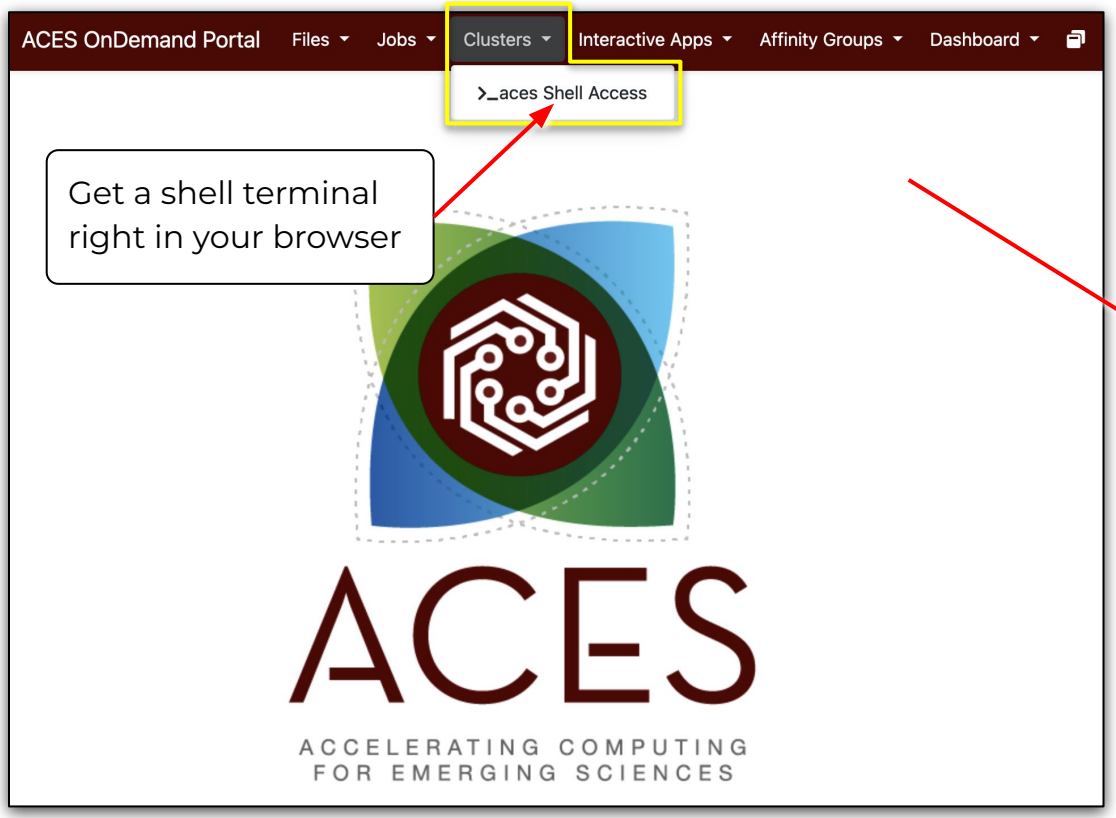
Anybody with an ACCESS ID can log in with ACCESS from anywhere

Tangent for Experts: Accessing Accelerators

- Example Slurm Scripts
- Memverge
- IPUs
- NEC Vector Engines

(see https://hprc.tamu.edu/kb/User-Guides/ACES/ for more)

Shell Access via the Portal



Job Script on ACES

```
#!/bin/bash
#NECESSARY JOB SPECIFICATIONS
#SBATCH --job-name=my_job
#SBATCH --time=2-00:00:00
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --cpus-per-task=96
#SBATCH --mem=488G
#SBATCH --partition=gpu
#SBATCH --gres=gpu:h100:2
#SBATCH --output=stdout.%x.%j
#SBATCH --error=stderr.%x.%j

# load required module(s)
module purge
module load GCC/13.1.0

./my_program.py
```

These parameters describe the resources needed for your program to the job scheduler

Choose your GPUs with --gres!

Script to execute
(In this case, set up environment and launch an executable)

Composable Job Script - FASTER

```
#!/bin/bash
#NECESSARY JOB SPECIFICATIONS
#SBATCH --job-name=my_job
#SBATCH --time=7-00:00:00
#SBATCH --ntasks-per-node=1
#SBATCH --cpus-per-task=64
#SBATCH --mem=250G
#SBATCH --partition=gpu
#SBATCH --gres=gpu:a100:4
#SBATCH --output=stdout.%x.%j
#SBATCH --error=stderr.%x.%j

# load required module(s)
module purge
module load GCC/13.1.0

./my_program.py
```

These parameters describe the resources needed for your program to the job scheduler

Choose your GPUs with --gres!

Script to execute
(In this case, set up environment and launch an executable)

Accessing MemVerge

```
srun --partition=memverge --time=24:00:00 --pty bash
```

Sample job file:

```
#!/bin/bash

##NECESSARY JOB SPECIFICATIONS
#SBATCH --job-name=Example           #Set the job name to Example
#SBATCH --time=24:00:00              #Set the wall clock limit to 24 hrs
#SBATCH --nodes=1                    #Request 1 nodes
#SBATCH --ntasks-per-node=64         #Request 64 tasks/cores per node
#SBATCH --mem=248G                   #Request 248G (248GB) per node
#SBATCH --output=Example.%j          #Redirect stdout/err to file
#SBATCH --partition=memverge         #Specify the MemVerge partition

#lines required to setup the environment for your code
# add the mm command in front of your executable to run with memory machine
mm executable
```

hprc.tamu.edu/kb/User-Guides/ACES/memverge

(Similarly, there are partitions for the PVCs, FPGAs, and Nextsilicon coprocessors)

Graphcore IPUs

- SSH into poplar from ACES
  - **`[username@login ~]$ ssh poplar1`**
- Enable the SDK environment
  - **`source /opt/gc/poplar/poplar_sdk-ubuntu_20_04-3.1.0+1205-58b501c780/poplar-ubuntu_20_04-3.1.0+6824-9c103dc348/enable.sh`**
  - **`mkdir -p /localdata/$USER/tmp`**
  - **`export TF_POPLAR_FLAGS=--executable_cache_path=/localdata/$USER/tmp`**
  - **`export POPTORCH_CACHE_DIR=/localdata/$USER/tmp`**
- Type **`gc-monitor`** to view the status of the IPUs:

Contact us first to be given access to poplar

```
mouse@poplar1:~$ gc-monitor
+-------------+-----------------------------------------------------------------------+
| gc-monitor  |              Partition: p16 [active] has 16 reconfigurable IPUs       |
+-------------+-----------------+--------+--------------+----------+---------+-----+-------+-------+
|    IPU-M    |      Serial     |IPU-M SW|Server version| ICU FW   | Type    | ID  | IPU#  |Routing|
+-------------+-----------------+--------+--------------+----------+---------+-----+-------+-------+
| 10.1.5.1    | 0010.0002.8213921 |      |    1.9.0     | 2.4.4    | M2000   | 0   |  3    |  DNC  |
| 10.1.5.1    | 0010.0002.8213921 |      |    1.9.0     | 2.4.4    | M2000   | 1   |  2    |  DNC  |
| 10.1.5.1    | 0010.0001.8213921 |      |    1.9.0     | 2.4.4    | M2000   | 2   |  1    |  DNC  |
| 10.1.5.1    | 0010.0001.8213921 |      |    1.9.0     | 2.4.4    | M2000   | 3   |  0    |  DNC  |
+-------------+-----------------+--------+--------------+----------+---------+-----+-------+-------+
| 10.1.5.2    | 0030.0002.8213921 |      |    1.9.0     | 2.4.4    | M2000   | 4   |  3    |  DNC  |
| 10.1.5.2    | 0030.0002.8213921 |      |    1.9.0     | 2.4.4    | M2000   | 5   |  2    |  DNC  |
| 10.1.5.2    | 0030.0001.8213921 |      |    1.9.0     | 2.4.4    | M2000   | 6   |  1    |  DNC  |
```

See also our Youtube channel for previous IPU training sessions!

hprc.tamu.edu/kb/User-Guides/ACES/Graphcore_IPUs

NEC Vector Engines

Access VE node

Environment setup

Turn on printing usage details

Run on 1 VE card (VE card 0) with 8 VE cores

Run on 2 VE cards (VE card 0 and 1) with 8 VE cores per card

Run on 8 VE cards (VE cards 0-7) with 8 VE cores per card

```
>$ ssh dss
>$ export PATH=/opt/nec/ve/bin/:$PATH
>$ source /opt/nec/ve/mpi/3.4.0/bin64/necmpivars.sh
>$ export VE_PROGINF=DETAIL
>$ mpirun -ve 0 -np 8 executable
>$ mpirun -ve 0-1 -np 16 executable
>$ mpirun -ve 0-7 -np 64 executable
```

Links to further documentation at
https://hprc.tamu.edu/kb/User-Guides/ACES/vectorengine/

Portal and Cluster Basics

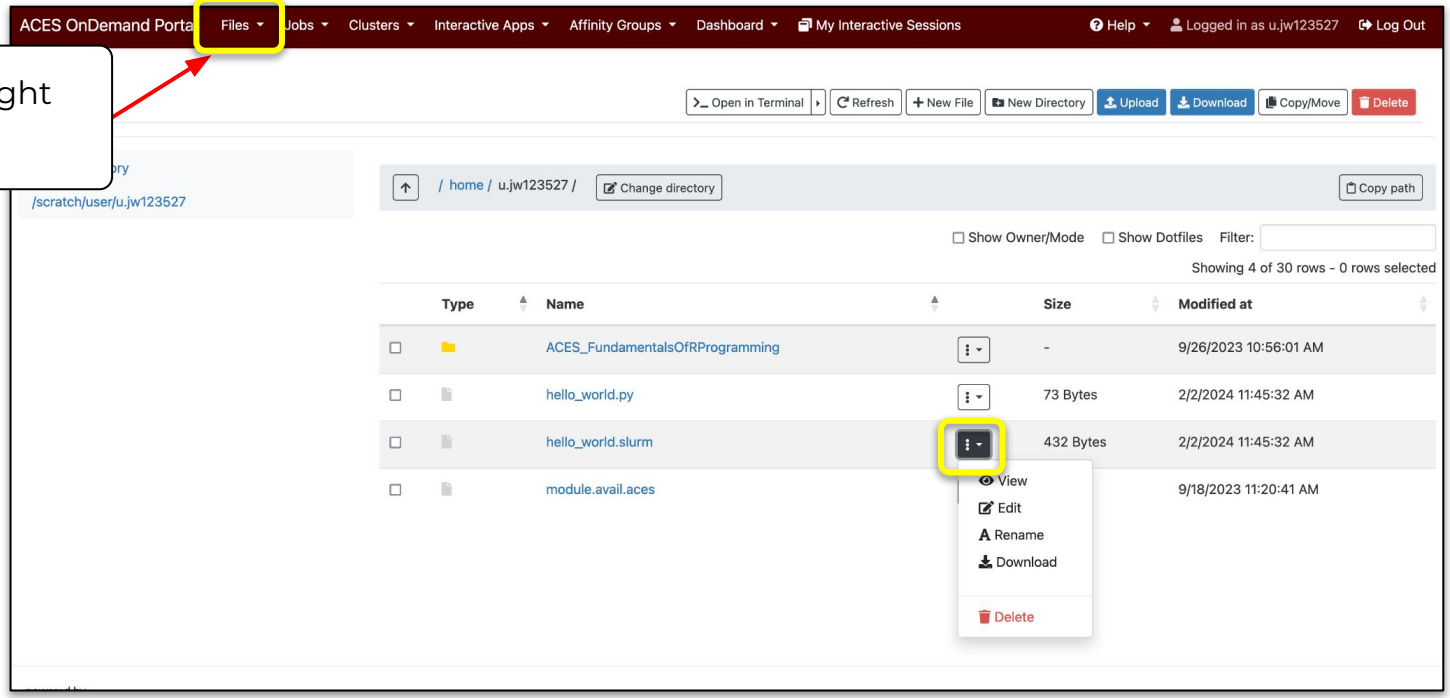
HPRC Portal Overview

At the top of the portal are several tabs

We'll look at each of them briefly

File Browsing, Viewing, and Editing

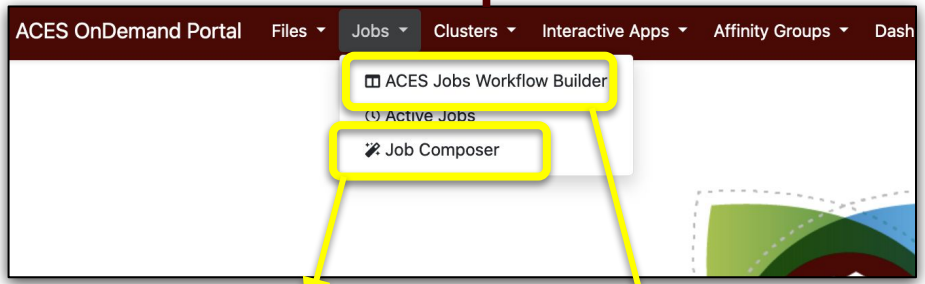
Manage your files right in your browser

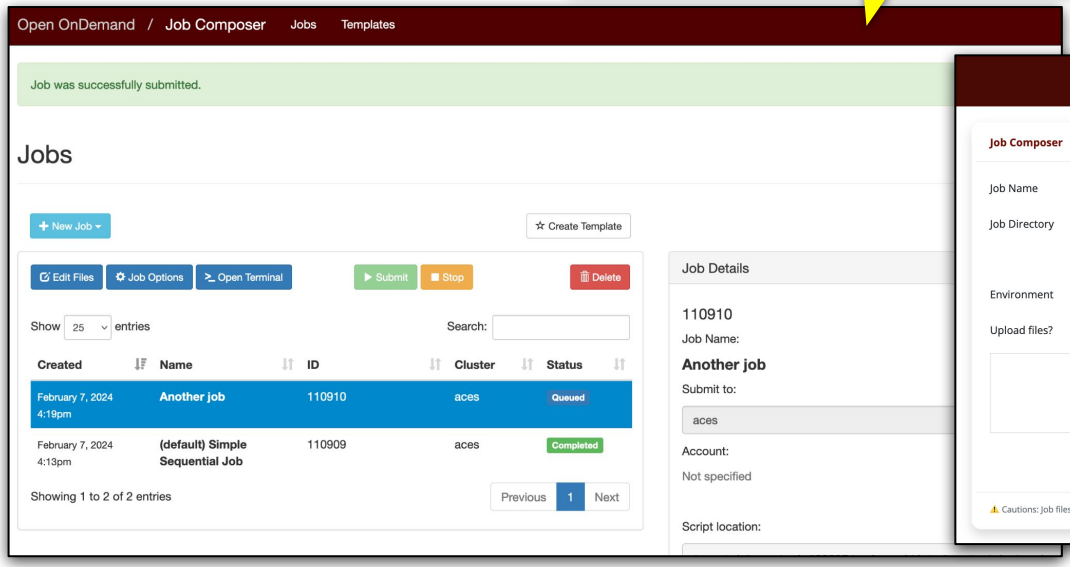


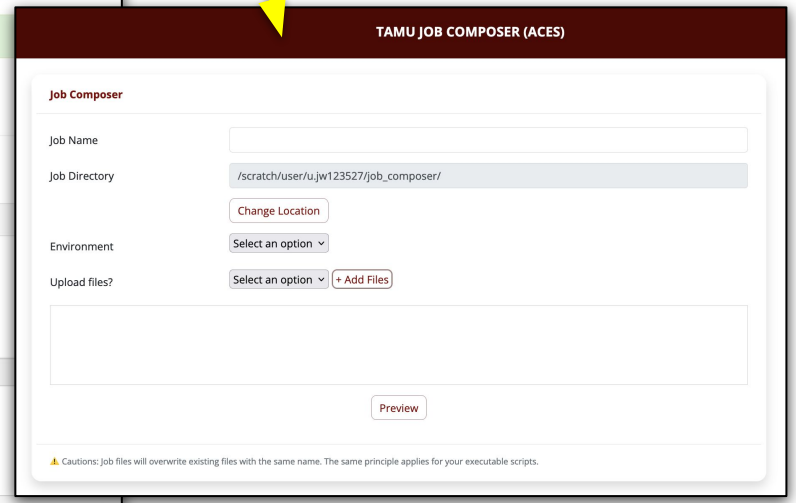
Portal Job Composer

Jobs tab:

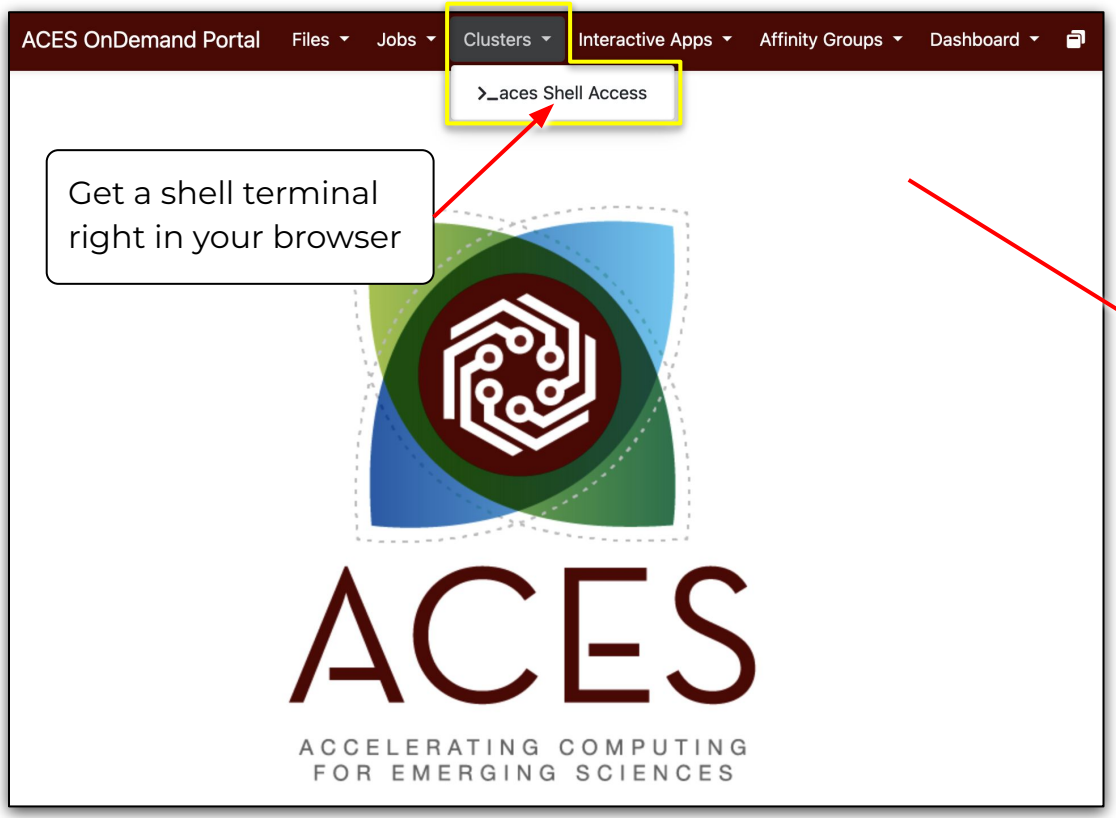
- Show status of jobs on clusters
- Write and submit jobs



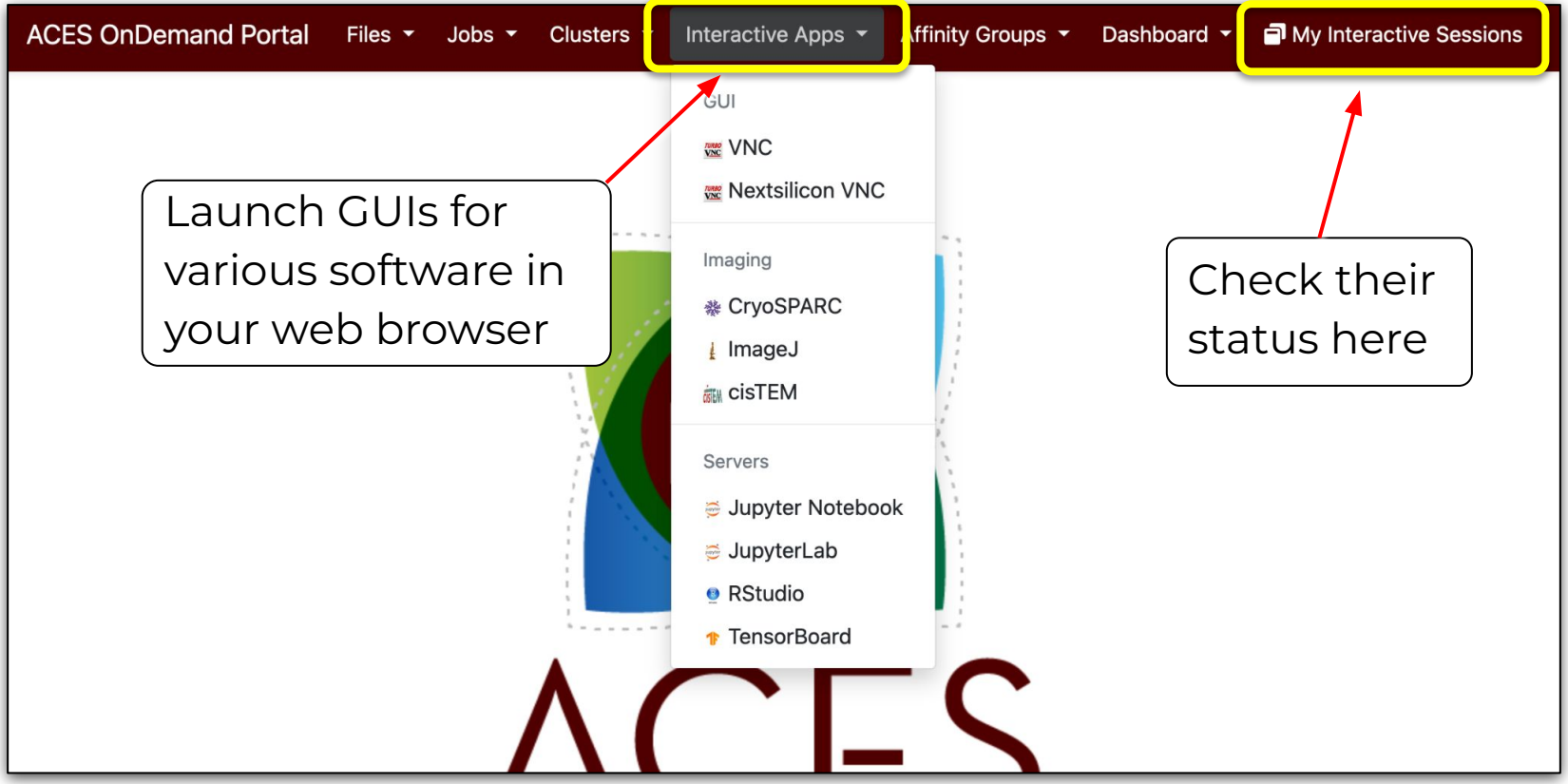




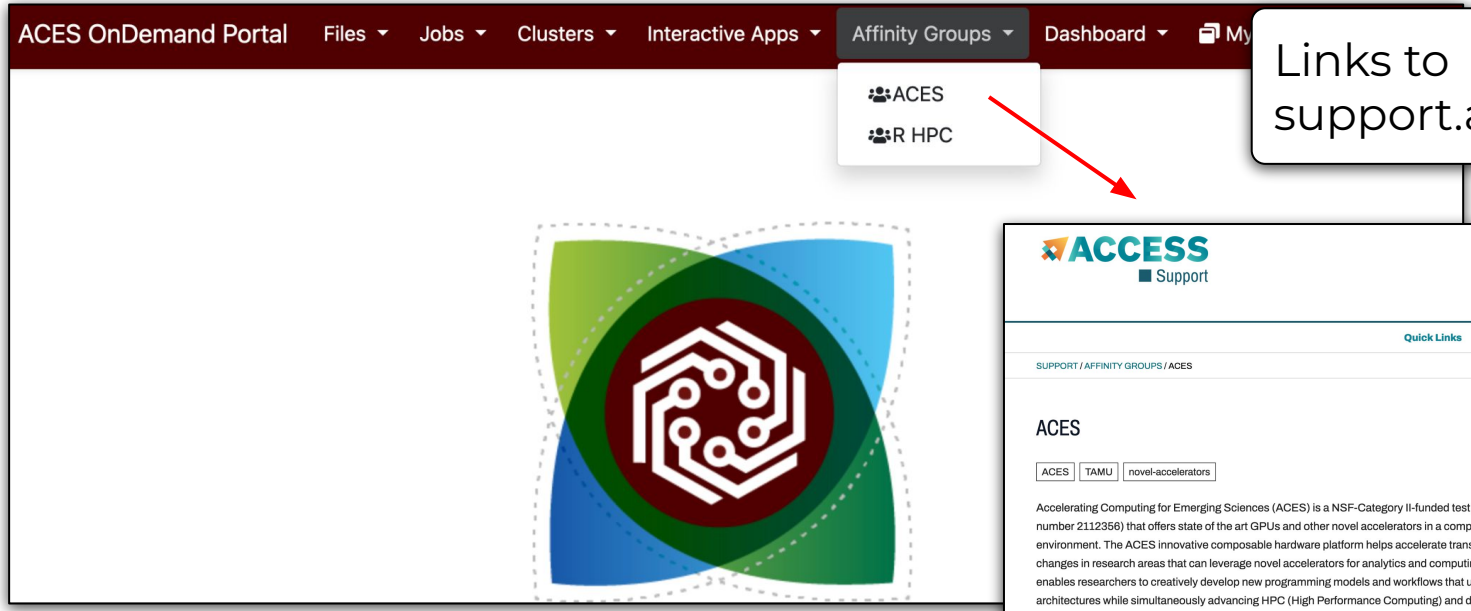
Shell Access via the Portal



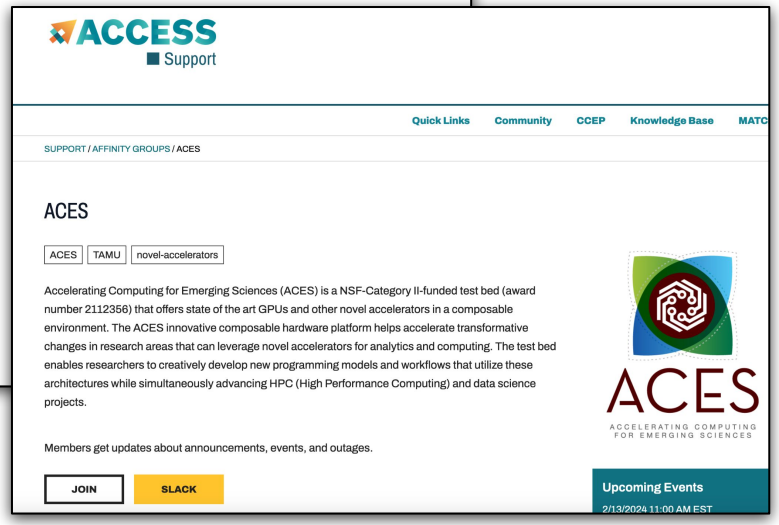
Interactive Apps



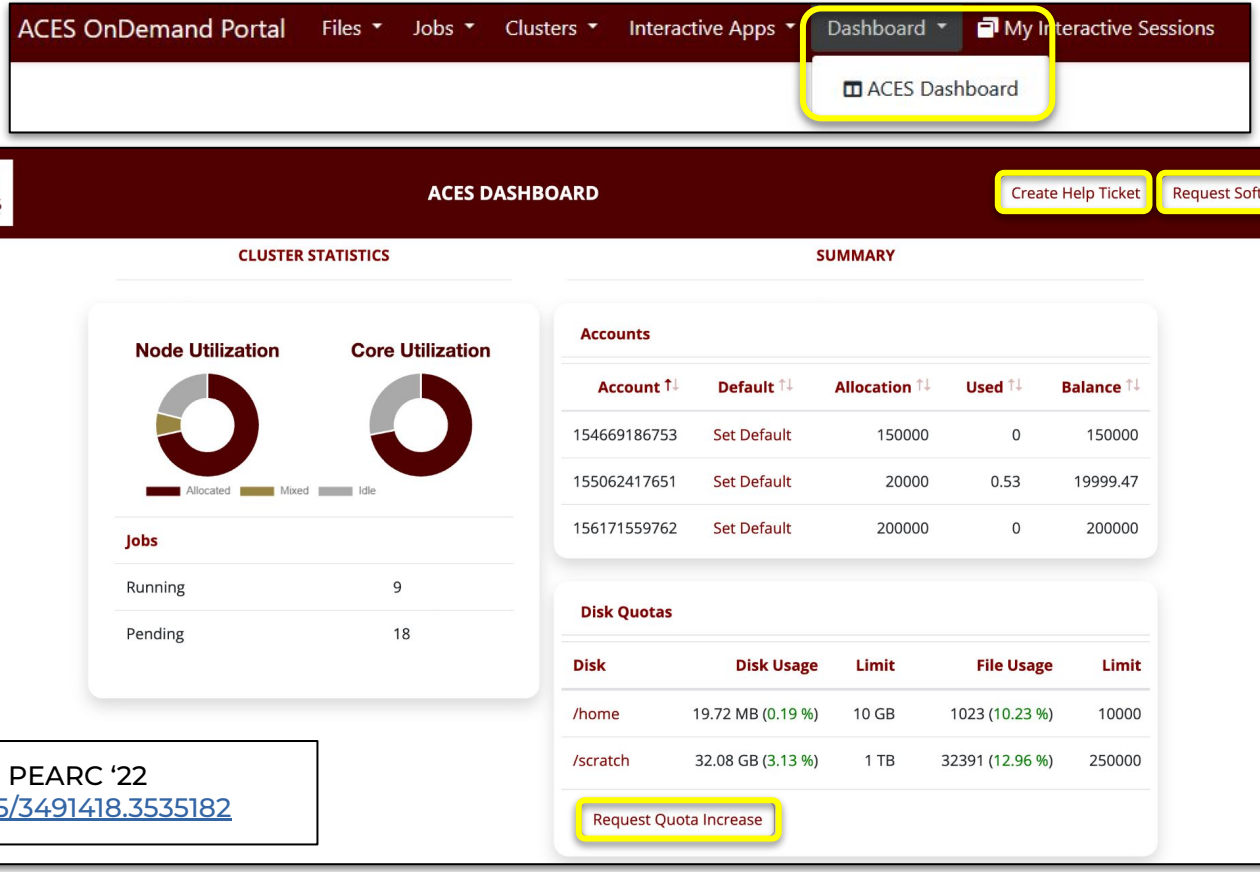
Join Affinity Groups



Links to support.access-ci.org



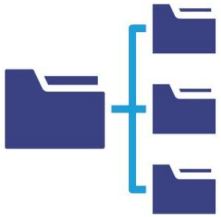
ACES Dashboard

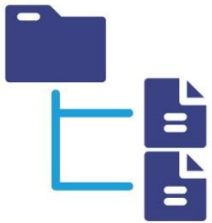


D. Pham et al. (2022) PEARC '22
https://doi.org/10.1145/3491418.3535182

File Systems and User Directories

Directory	Environment Variable	Space Limit	File Limit	Intended Use
/home/$USER	$HOME	10 GB	10,000	Small to modest amounts of processing.
/scratch/user/$USER	$SCRATCH	1 TB	250,000	Temporary storage of large files for on-going computations. Not intended to be a long-term storage area.
/scratch/group/PROJECTID	$PROJECT	5 TB	500,000	High performance storage for specific storage allocation requests. Not purged while allocation is active. Available upon request.





Service Units

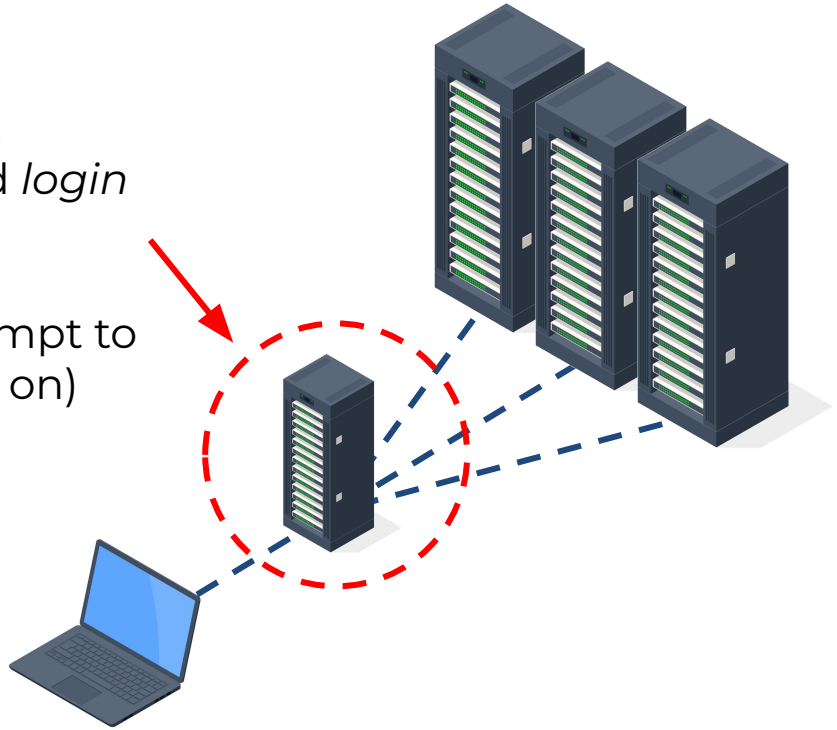
- Service Units (SUs) are part of our account management system (AMS).
- 1 SU ~ 1 core-hour on a CPU … more if using accelerators or extra memory
- With a default allocation (not available on ACES), you start with 20,000 SUs
- SUs are charged as your jobs spend time on the compute nodes (we'll see how later).
  (Work on login nodes is not charged, but you can't do big computing there!)
- You can check your SU balance on both:
  - The command line
  - The HPRC Portal

https://hprc.tamu.edu/kb/User-Guides/AMS/Service_Unit/
https://hprc.tamu.edu/kb/User-Guides/AMS/UI/
https://hprc.tamu.edu/policies/allocations.html

Logging-In

When you first log in, you're on a dedicated *login node*.

(check your shell prompt to see which one you're on)



Login nodes are not for running big processes!
There are rules:

- No processes longer than 1 hr
- Sessions idle for 1 hr will be killed
- Don't use more than 8 cores
- Don't use "`sudo`"

Command-line Tools: Quota Usage

Check your file and storage use with the “showquota” command:

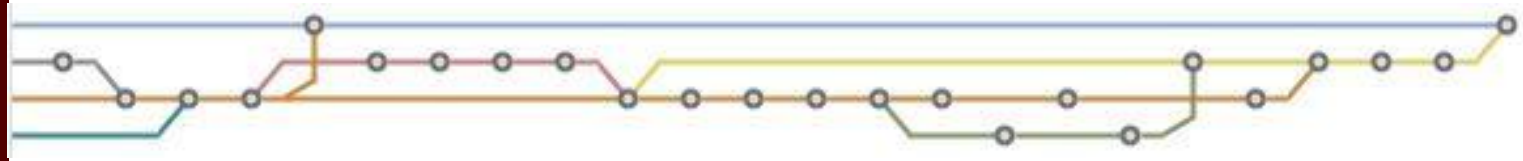
```
[username@aces ~]$ showquota

Your current disk quotas are:
Disk                    Disk Usage      Limit   File Usage   Limit
/home/username               1.4G      10.0G         3661   10000
/scratch/user/username     117.6G       1.0T        24226  250000
/scratch/group/projectid   510.5G       5.0T       128523  500000
```

Command-line Tools : Allocation Usage

Check your allocation balances with the “myproject” command:

```
[username@aces ~]$ myproject
 ================================================================================================
 List of user's Project Accounts
 ------------------------------------------------------------------------------------------------
 |  Account    | Default | Allocation |Used & Pending SUs|   Balance  |          PI        |
 ------------------------------------------------------------------------------------------------
 |1228000223136|        N|   150000.00|              0.00|    10000.00| Last, First        |
 ------------------------------------------------------------------------------------------------
 |1428000243716|        Y|    20000.00|              0.00|    20000.00| Last, First        |
 ------------------------------------------------------------------------------------------------
 |1258000247058|        N|     5000.00|              0.00|     5000.00| Last, First        |
 ------------------------------------------------------------------------------------------------
```

Software Infrastructure



2,000+ Software Modules on FASTER!

SOFTWARE MODULES ON THE FASTER CLUSTER

ACES Software Modules | FASTER Software Modules | Grace Software Modules | Terra Software Modules

Last Updated: Aug 7 11:58:37 CDT

The available software for the FASTER cluster is listed in the table. Click on any software package name to get more information such as the available versions, additional documentation if available, etc.

Show 10 entries Search: tensorflow

Name	Description
einops	'Flexible and powerful tensor operations for readable and reliable code. Supports numpy, pytorch, tensorflow, jax, and others.'
Horovod	'Horovod is a distributed training framework for TensorFlow.'
Keras	'Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. '
ONNX-Runtime	'ONNX Runtime inference can enable faster customer experiences and lower costs, supporting models from deep learning frameworks such as PyTorch and TensorFlow/Keras as well as classical machine learning libraries such as scikit-learn, LightGBM, XGBoost, etc. ONNX Runtime is compatible with different hardware, drivers, and operating systems, and provides optimal performance by leveraging hardware accelerators where applicable alongside graph optimizations and transforms.'
segmentation-models	'Python library with Neural Networks for Image Segmentation based on Keras and TensorFlow.'
tensorboard	'TensorBoard is a suite of web applications for inspecting and understanding your TensorFlow runs and graphs.'
TensorFlow	'An open-source software library for Machine Intelligence'

hprc.tamu.edu/software/faster

1,300+ Software Modules on ACES!

SOFTWARE MODULES ON THE ACES CLUSTER

ACES Software Modules | FASTER Software Modules | Grace Software Modules | Terra Software Modules

Last Updated: Aug 7 12:04:43 CDT

The available software for the ACES cluster is listed in the table. Click on any software package name to get more information such as the available versions, additional documentation if available, etc.

Show 10 entries Search: TensorFlow

Name	Description
CliMetLab	'CliMetLab is a Python package aiming at simplifying access to climate and meteorological datasets, allowing users to focus on science instead of technical issues such as data access and data formats. It is mostly intended to be used in Jupyter notebooks, and be interoperable with all popular data analytic packages, such as Numpy, Pandas, Xarray, SciPy, Matplotlib, etc. as well as machine learning frameworks, such as Tensorflow, Keras or PyTorch.'
einops	'Flexible and powerful tensor operations for readable and reliable code. Supports numpy, pytorch, tensorflow, jax, and others.'
Horovod	'Horovod is a distributed training framework for TensorFlow.'
tensorboard	'TensorBoard is a suite of web applications for inspecting and unders

hprc.tamu.edu/software/aces

Installing Software

- Software pages for instructions and examples:
  - https://hprc.tamu.edu/kb/Software
  - hprc.tamu.edu/software
- Researchers can install software in their home/scratch directories
  - Do NOT run the "sudo" command when installing software
  - Watch your file quotas! Install in $SCRATCH!
- Contact us if you need help or want to request software
  - We can install software cluster-wide
  - Requests can be sent from the Dashboard
- License-restricted software
  - Check on command line with `license_status`
  - Contact help@hprc.tamu.edu

Computing Environment

Managing software versions using lmod

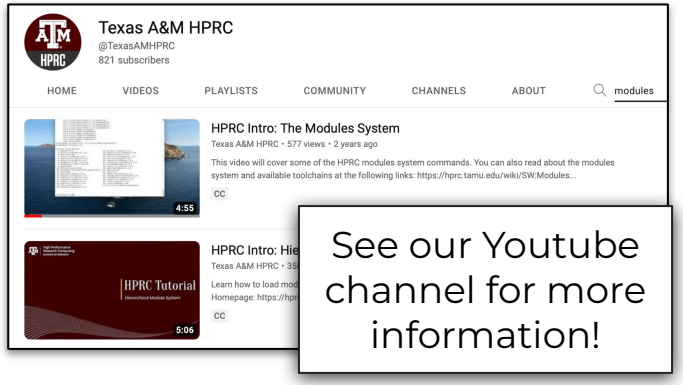
- Uses the command: **module**
- Each version of a software, application, library, etc. is available as a module.
  - Module names have the format:

toolchain-name / version toolchain-name / version

GCC/10.3.0 OpenMPI/4.1.1

software-name / version

PyTorch/1.10.0



See our Youtube channel for more information!

- module sets the correct environment variables for you.

hprc.tamu.edu/kb/Software/useful-tools/Modules

Module Usage Basics

`module avail`

- Lists all available modules (may be slow).
- Navigation:
  -spacebar, arrows, or `j` and `k`
  -quit with `q`
- Case-sensitive search: `/`
- Use **`mla`** instead to save results to a file as well

`module spider <word>`

- Case-sensitive search for modules with “word” in name.
- Provide an exact name to see dependencies.

```
[u.jw123527@aces-login2 ~]$ module spider Python

----------------------------------------------------------------------------------------------------------------------
  Python:
----------------------------------------------------------------------------------------------------------------------
    Description:
      Python is a programming language that lets you work more quickly and integrate your systems more effectively.

     Versions:
        Python/2.7.18-bare
        Python/2.7.18
        Python/3.8.6
        Python/3.9.5-bare
        Python/3.9.5
        Python/3.9.6-bare
        Python/3.9.6
        Python/3.10.4-bare
        Python/3.10.4
        Python/3.10.8-bare
        Python/3.10.8
        Python/3.11.3
        Python/3.11.5
     Other possible modules matches:
        Biopython  Boost.Python  Brotli-python  IPython  LASSO-Python  Python-bundle-PyPI  flatbuffers-python  graphvi

----------------------------------------------------------------------------------------------------------------------
  To find other possible module matches execute:

      $ module -r spider '.*Python.*'
```

Module Usage Basics

`module load <module>`

- add <module> paths to the current environment variables

`module list`

- See what modules are loaded in your current session

`module purge`

- Unload all modules

```
[u.jw123527@aces-login2 ~]$ module load GCCcore/13.2.0
[u.jw123527@aces-login2 ~]$ module list

Currently Loaded Modules:
  1) GCCcore/13.2.0



[u.jw123527@aces-login2 ~]$ module purge
[u.jw123527@aces-login2 ~]$ module list
No modules loaded
[u.jw123527@aces-login2 ~]$
```

Sample Job Script Structure

```
#!/bin/bash

##NECESSARY JOB SPECIFICATIONS
#SBATCH --job-name=hello_world
#SBATCH --time=00:15:00
#SBATCH --ntasks=2
#SBATCH --ntasks-per-node=2
#SBATCH --nodes=1
#SBATCH --mem=3G
#SBATCH --output=hello_world_log.%j

# load required module(s)
module purge
module load GCCcore/11.3.0
module load Python/3.10.4
python hello_world.py

# Job Environment variables
echo $SLURM_JOBID
echo $SLURM_SUBMIT_DIR
echo $TMPDIR
echo $SCRATCH
```

This is a single-line comment and not run as part of the script.

These parameters describe your job to the job scheduler.
The lines starting with #SBATCH are NOT comments!
See the Knowledge Base for more info.

Whatever commands or scripts you want to run.
Here, we set up the modules we need for our environment, run a python program, and print out some environment variables.

Submit a Job and Check Job Status

Submit job

```
sbatch example01.job
```

```
Submitted batch job 6853258
(from job_submit) your job is charged as below
              Project Account: 122792016265
              Account Balance: 1687.066160
              Requested SUs:   3
```

Check status

```
squeue -u $USER
```

or

```
squeue --me
```

```
JOBID    NAME      USER       PARTITION  NODES  CPUS  STATE    TIME         TIME_LEFT  START_TIME          REASON     NODELIST
6853258  jobname   someuser   cpu        2      96    RUNNING  3-07:36:50   16:23:10   2023-01-23T17:27:3  None       ac[180,202]
6853257  jobname   someuser   cpu        2      96    RUNNING  3-07:36:56   16:23:04   2023-01-23T17:27:2  None       ac[523-524]
```

Checking Job Resources

Useful commands:

- `sinfo`: check status of batch queues
- `pestat`: check node status
- `gpuavail` : check gpu availability
- `maxconfig`: check SU usage of jobs

sinfo : Current Queues

```
FASTER:
PARTITION          AVAIL  TIMELIMIT    JOB_SIZE    NODES(A/I/O/T)    CPUS(A/I/O/T)
cpu*               up     7-00:00:00   1-32        36/59/27/122      1526/4554/1728/7808
gpu                up     7-00:00:00   1-32        15/26/11/52       478/2146/704/3328

ACES:
PARTITION          AVAIL  TIMELIMIT    JOB_SIZE    NODES(A/I/O/T)    CPUS(A/I/O/T)
cpu*               up     7-00:00:00   1-64        17/3/36/56        1540/380/3456/5376
gpu                up     2-00:00:00   1-15        0/0/16/16         0/0/1536/1536
pvc                up     2-00:00:00   1-15        0/0/20/20         0/0/1920/1920
bittware           up     2-00:00:00   1           0/2/0/2           0/192/0/192
d5005              up     2-00:00:00   1           0/2/0/2           0/192/0/192
memverge           up     2-00:00:00   1           0/6/2/8           0/576/192/768
nextsilicon        up     2-00:00:00   1           0/2/0/2           0/192/0/192
gpu-hybrid         up     2-00:00:00   1-infinite  0/0/35/35         0/0/3360/3360
```

For the NODES and CPUS columns:

- **A = Active (in use by running jobs)**
- **I = Idle (available for jobs)**
- **O = Offline (unavailable for jobs)**
- **T = Total**

FASTER Charging Scheme

A Service Unit (SU) on **FASTER** is equivalent to one core or **3.5** GB memory usage for one hour– plus additional SUs for GPU usage.

Number of Cores	GB of memory per core	Total Memory (GB)	Hours	SUs charged
1	3	3	1	1
1	4	4	1	2
1	250	250	1	64
64	250	250	1	64

On FASTER each T4 GPU would be an additional 64 SUs for one hour and each A100, A10, A30, or A40 would be an additional 128 SUs for one hour.

hprc.tamu.edu/kb/User-Guides/AMS/Service_Unit

ACES: Charging Scheme

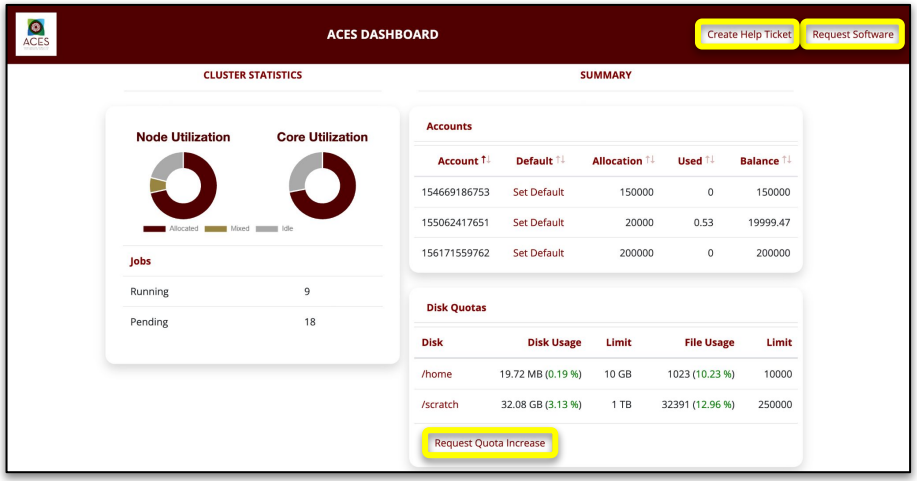
Resource	Service Units (per hour)	ACCESS Credits (per hour)
Intel SPR / Ice Lake	1	0.1
NVIDIA H100	240	30
NVIDIA A30 and Intel PVC GPUs	120	15
Bittware Agilex FPGA	200	25
Intel D5005 FPGA	50	6.2
NEC Vector Engine	150	18.7
NextSilicon coprocessor	100	12.5
Graphcore IPU Classic	120	15
Graphcore IPU Bow	150	18.7
Intel Optane Memory	150	18.7

Need Help?

First check the FAQ: https://hprc.tamu.edu/kb/FAQ/Accounts

- ACES user Guide: https://hprc.tamu.edu/kb/User-Guides/ACES
- FASTER user Guide: https://hprc.tamu.edu/kb/User-Guides/FASTER
- Email your questions to help@hprc.tamu.edu

Remember the Dashboard!



Need Help?

Help us help you -- tell us:

- Which cluster
- Username
- Job id(s) if any
- Location of your jobfile, input/output files
- Application used if any
- Module(s) loaded if any
- Error messages
- Steps you have taken, so we can reproduce the problem

High Performance
Research Computing
DIVISION OF RESEARCH

Thank you

Questions?