

ACES: Software for AI on HPC

February 18, 2025

Zhenhua He & Richard Lawrence



High Performance
Research Computing

DIVISION OF RESEARCH



High Performance Research Computing | hprc.tamu.edu | NSF Award #2112356

High Performance Computing

High Performance Computing (HPC): use supercomputers or clusters of high performance machines to solve complex computational problems.

HPC features:

- Parallel Computing
- Specialized Hardware
- Scalable Architecture

Why HPC Important for AI

AI/ML involves **computationally intensive** tasks such as training large models, handling massive datasets, and performing real-time inference.

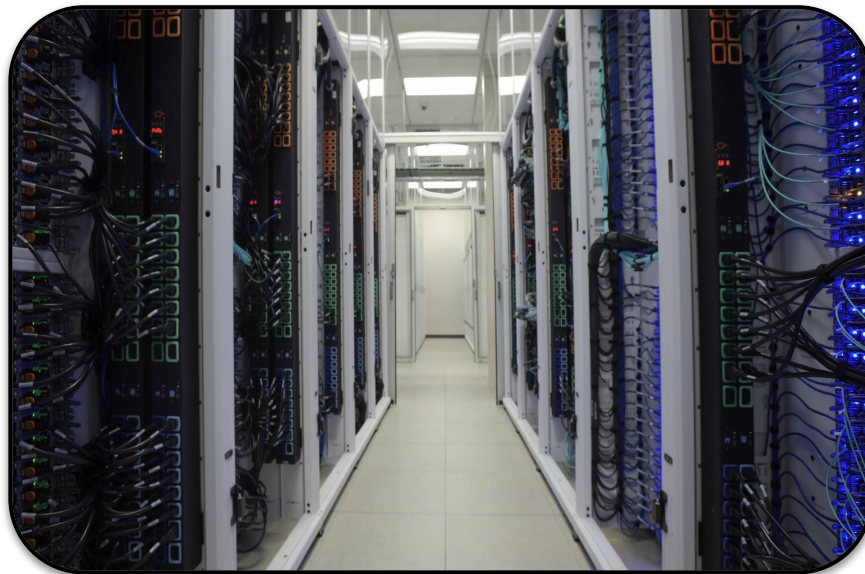
HPC provides the **necessary infrastructure** to:

- Accelerate AI Training and inference: Accelerators such as GPUs, IPUs, etc
- Handle Large Datasets: High-speed storage and interconnects (e.g., InfiniBand)
- Optimize Resource Utilization: Slurm workload manager
- Enable Research at Scale
- ...

Overview of HPRC Resources

We operate five clusters

- Grace
- ViDaL
- Launch
- **FASTER**
- **ACES**

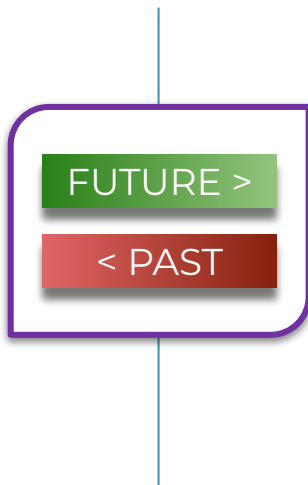


See hprc.tamu.edu/resources
for more info on all our clusters

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

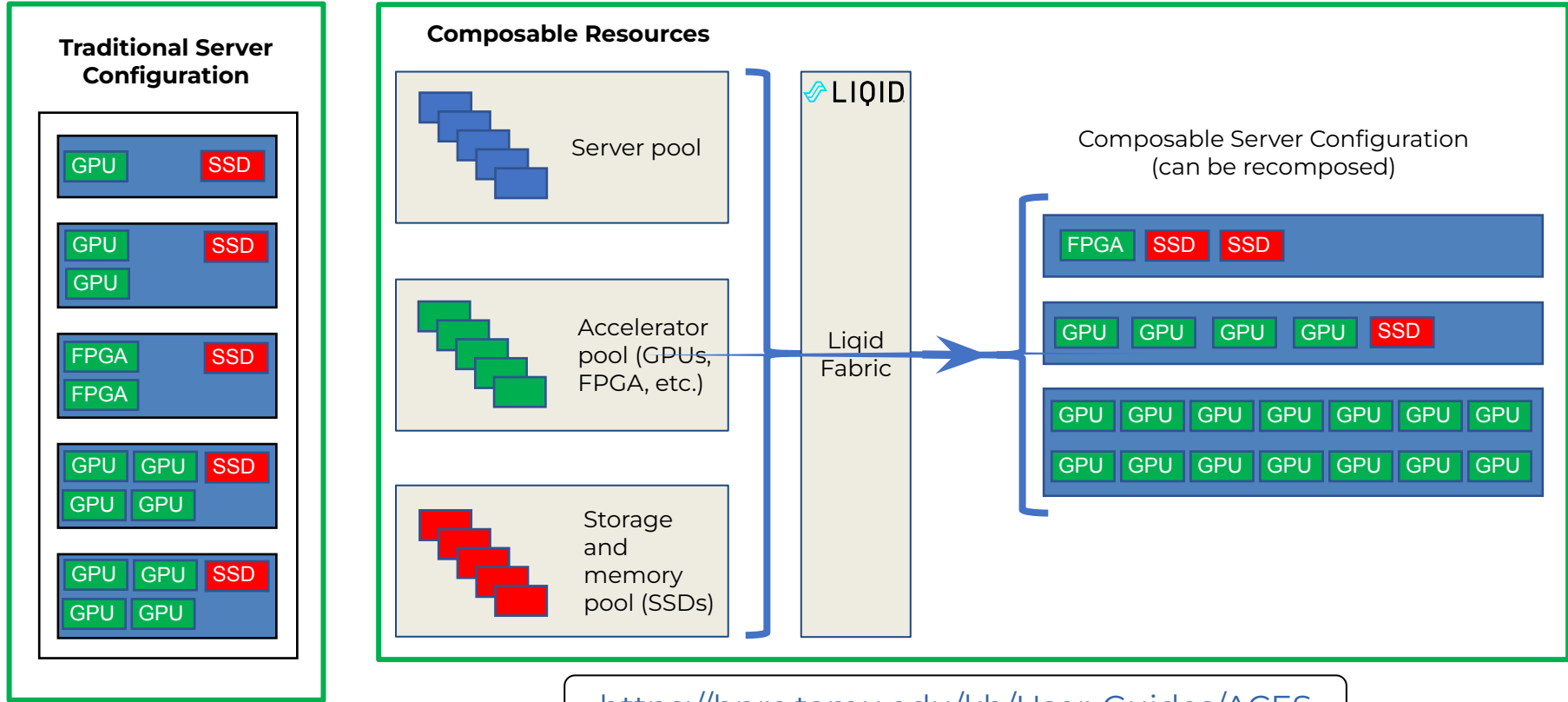


HPC for AI

- 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

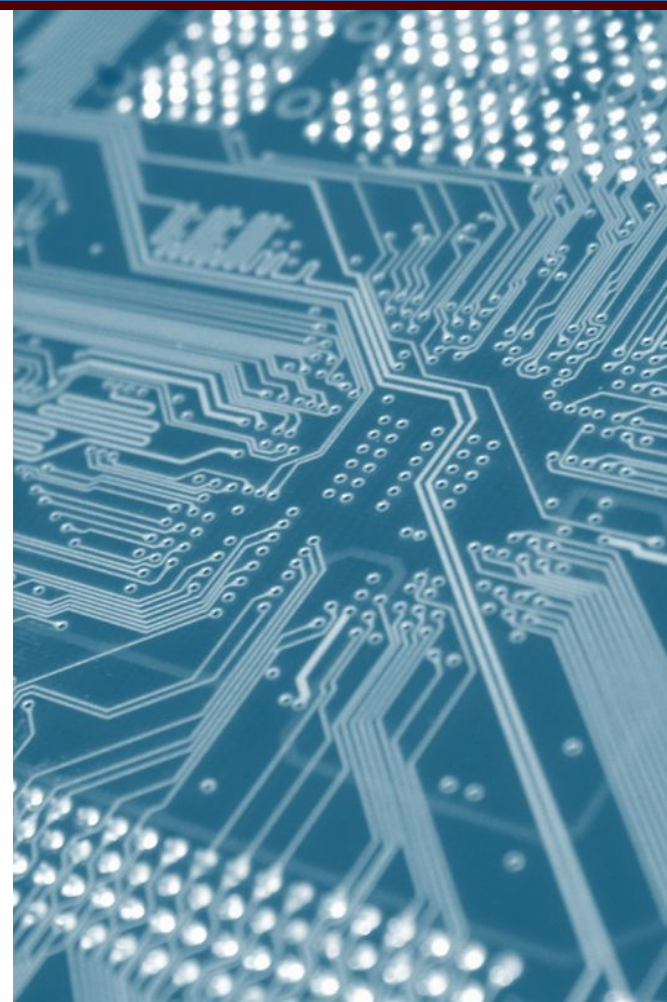


<https://hprc.tamu.edu/kb/User-Guides/ACES>

HPRC's Composable Clusters

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

Focusing on ACES today



Status of Composability

Composability is currently handled by system administrators

- Contact help@hprc.tamu.edu to request a given composable configuration for a cluster.
- Once the composed node has been created, simply specify the relevant resources in your Slurm job file.
- In-progress: making Slurm and Liquid handle composable configurations automatically
- *Note:* The IPU's and NEC Vector Engines cannot be composed.

ACES Overview

NSF ACES

Accelerating Computing for Emerging Sciences

Our Mission:

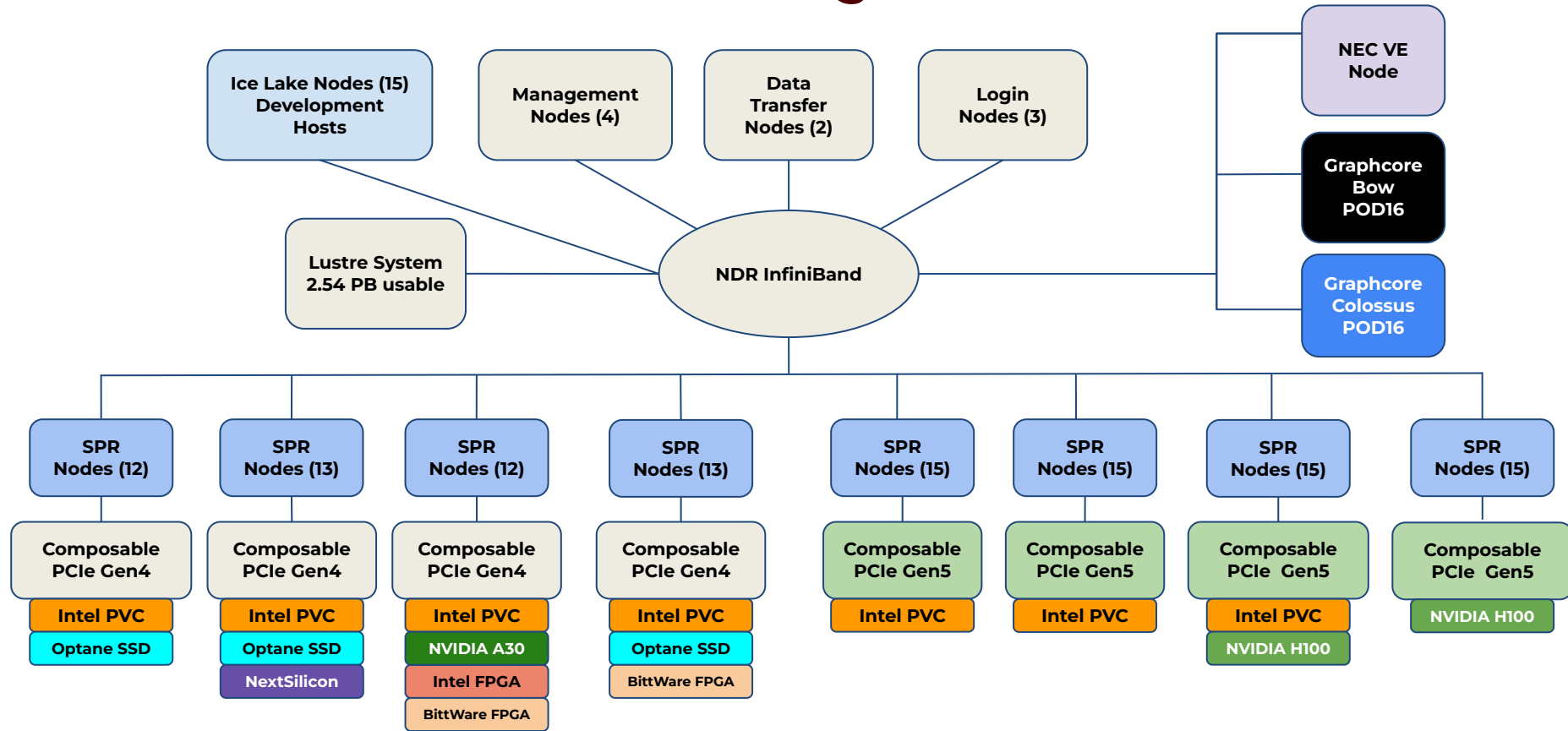
- NSF ACSS CI testbed
- Offer an accelerator testbed for numerical simulations and **AI/ML workloads**
- Provide consulting, technical guidance, and training to researchers
- Collaborate on computational and data-enabled research



ACES In Action



ACES Configuration



ACES System Description

Component	Quantity	Description
Sapphire Rapids Nodes: Compute Nodes Data Transfer Nodes Login & Management Nodes	110 nodes 2 nodes 5 nodes	96 cores per node, dual Intel Xeon 8468 processors 512 GB DDR5 memory 1.6 TB NVMe storage Compute: NVIDIA Mellanox NDR 200 Gbps InfiniBand adapter DTNs & Login & Management nodes: 100 Gbps Ethernet adapter
Ice Lake Login & Management Nodes	2 nodes	64 cores per node, dual Intel Xeon 8352Y processors 512 GB DDR4 memory 1.6 TB NVMe storage NVIDIA Mellanox NDR 200 Gbps InfiniBand adapter
PCIe Gen4 Composable Infrastructure	50 SPR nodes	Dynamically reconfigurable infrastructure that allows up to 20 PCIe cards (GPU, FPGA, etc.) per compute node
PCIe Gen5 Composable Infrastructure	60 SPR nodes	Dynamically reconfigurable infrastructure that allows up to 16 H100s or 14 PVCs per compute node
NVIDIA InfiniBand (IB) Interconnect	110 nodes	Two leaf and two spine switches in a 2:1 fat tree topology
DDN Lustre Storage	2.5 PB usable	HDR IB connected flash and disk storage for Lustre file systems

ACES Accelerators/Components

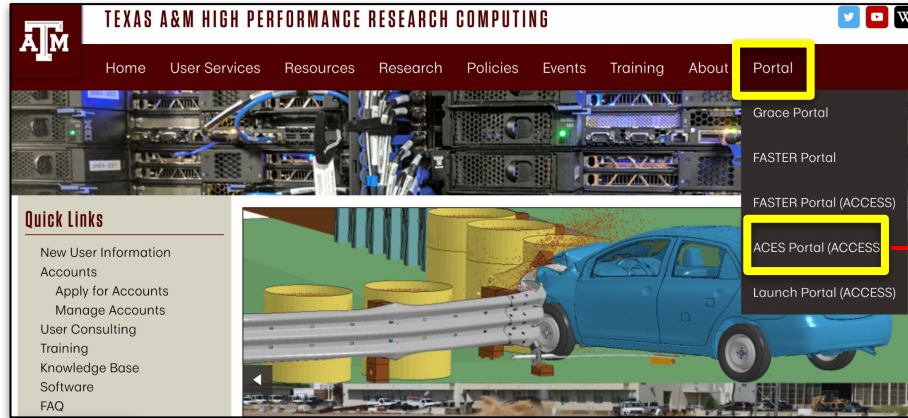
Component	Quantity	Description
Graphcore IPU	32	16 Colossus GC200 IPUs, 16 Bow IPUs. Each IPU group hosted with a CPU server as a POD16 on a 100 GbE RoCE fabric
<i>FPGAs:</i>		
Intel PAC D5005	2	Accelerator with Intel Stratix 10 GX FPGA and 32 GB DDR4
BittWare IA-840F	3	Accelerator with Agilex AGF027 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 SSDs addressable as memory w/ MemVerge Memory Machine.
<i>NVIDIA GPUs:</i>		
H100	30	For HPC, DL Training, AI Inference
A30	4	For AI Inference and Mainstream Compute
Intel PVC GPUs	120	Intel GPUs for HPC, DL Training, AI Inference

Refer to our Knowledge Base for more:

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

Access ACES

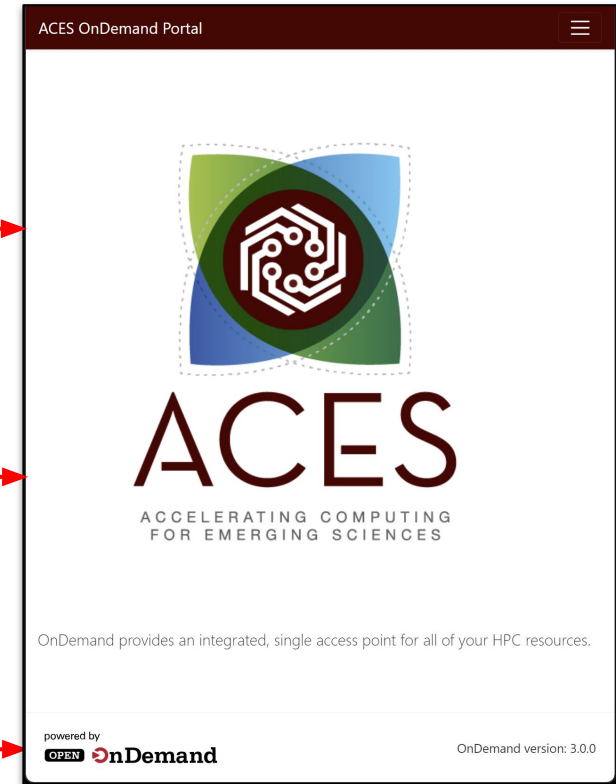
ACES Portal



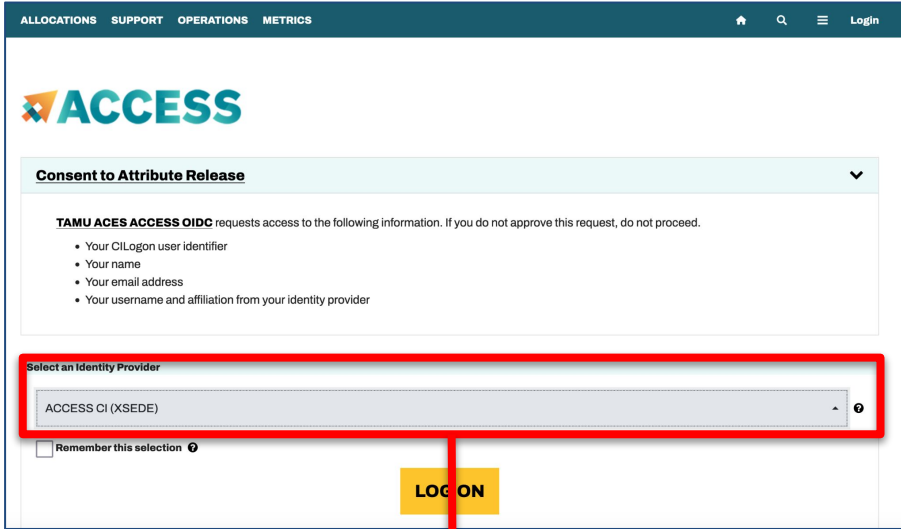
ACES Portal portal-aces.hprc.tamu.edu
is the web-based user interface for the ACES cluster

[HPRC Portal YouTube tutorials](#)

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

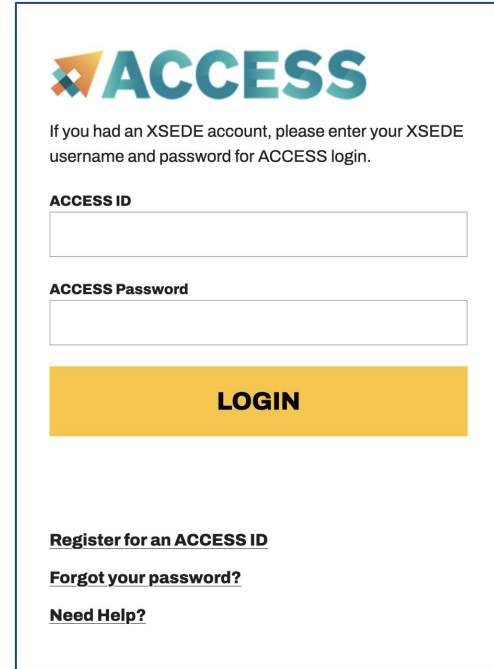


Accessing ACES via the Portal (ACCESS)



The screenshot shows the ACCESS portal interface. At the top is a navigation bar with links: ALLOCATIONS, SUPPORT, OPERATIONS, METRICS, and a Login link. Below the navigation bar is the ACCESS logo. A section titled "Consent to Attribute Release" contains a message from TAMU ACES ACCESS OIDC and a list of requested attributes: CILogon user identifier, name, email address, and username/affiliation. Below this is a "Select an Identity Provider" dropdown menu with "ACCESS CI (XSEDE)" selected. A red rectangle highlights this dropdown. Below the dropdown is a "Remember this selection" checkbox and a yellow "LOG ON" button. A red line points from the "LOG ON" button to the text below.

Select the Identity Provider appropriate for your account.

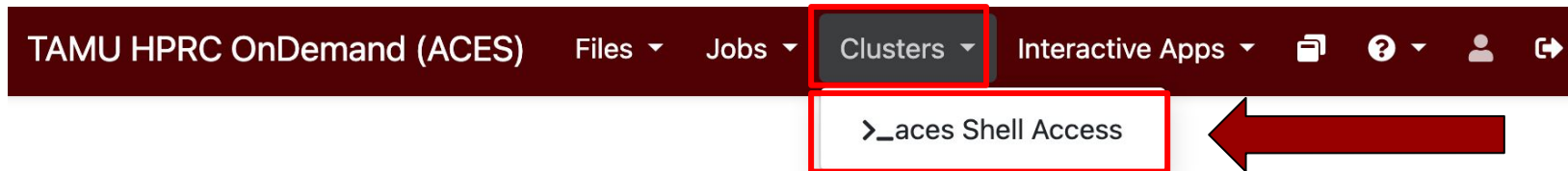


The screenshot shows the ACCESS portal login page. At the top is the ACCESS logo. Below it is a message: "If you had an XSEDE account, please enter your XSEDE username and password for ACCESS login." There are two input fields: "ACCESS ID" and "ACCESS Password". Below these fields is a yellow "LOGIN" button. At the bottom, there are links: "Register for an ACCESS ID", "Forgot your password?", and "Need Help?".

Log in using your ACCESS or institutional credentials.

Get a Shell on ACES

Click on “Clusters” menu → _aces Shell Access



Success!

Welcome to the ACES login node.

Check which login node you are on.

```
Host: login.aces Themes: Default
| Consulting:      help@hprc.tamu.edu (preferred) or (979) 845-0219 |
| ACES Documentation: https://hprc.tamu.edu/kb/User-Guides/ACES |
| FASTER Documentation: https://hprc.tamu.edu/kb/User-Guides/FASTER |
| Grace Documentation: https://hprc.tamu.edu/kb/User-Guides/Grace |
| Terra Documentation: https://hprc.tamu.edu/kb/User-Guides/Terra |
| YouTube Channel: https://www.youtube.com/texasamhprc |
|=====|

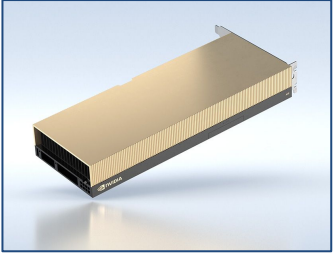
*****
*                               *
*      === IMPORTANT POLICY INFORMATION ===      *
* - Unauthorized use of HPRC resources is prohibited and subject to *
*   criminal prosecution. *
* - Use of HPRC resources in violation of United States export control *
*   laws and regulations is prohibited. Current HPRC staff members are *
*   US citizens and legal residents. *
* - Sharing HPRC account and password information is in violation of *
*   Texas State Law. Any shared accounts will be DISABLED. *
* - Authorized users must also adhere to ALL policies at: *
*   https://hprc.tamu.edu/policies/ *
*****

!! WARNING: THERE ARE ONLY NIGHTLY BACKUPS OF USER HOME DIRECTORIES. !!

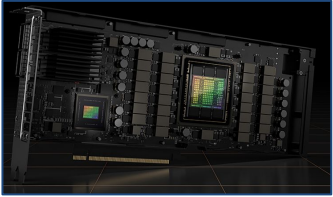
Please restrict usage to 8 CORES across ALL login nodes.
Users found in violation of this policy will be SUSPENDED.

To see these messages again, run the motd command.
Your current disk quotas are:
Disk                               Disk Usage      Limit      File Usage      Limit
/home/u.zh108696                   4.0G           10.0G        2361           10000
/scratch/user/u.zh108696          275.4G         1.0T        352057         1000000
Type 'showquota' to view these quotas again.
[u.zh108696@aces-login1 ~]$
```

NVIDIA GPUS on ACES



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

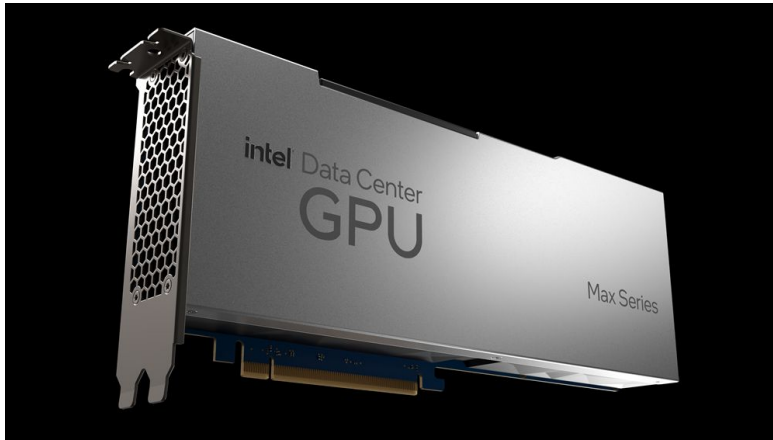


H100s Support computationally intensive workloads employing numerical simulations and AI/ML methods.

Specify in Slurm file:

```
#SBATCH --partition=gpu  
#SBATCH --gres=gpu:<gpu type>:<number>
```

Intel Data Center Max GPU 1100 GPUs (PVC GPUs)



PVC training event ([Link](#)) currently planned for March 4th!

See also our Youtube channel for previous PVC training sessions!

Intel GPUs for HPC, DL Training, AI Inference

Specify in Slurm file:

```
#SBATCH --partition=pvc
```

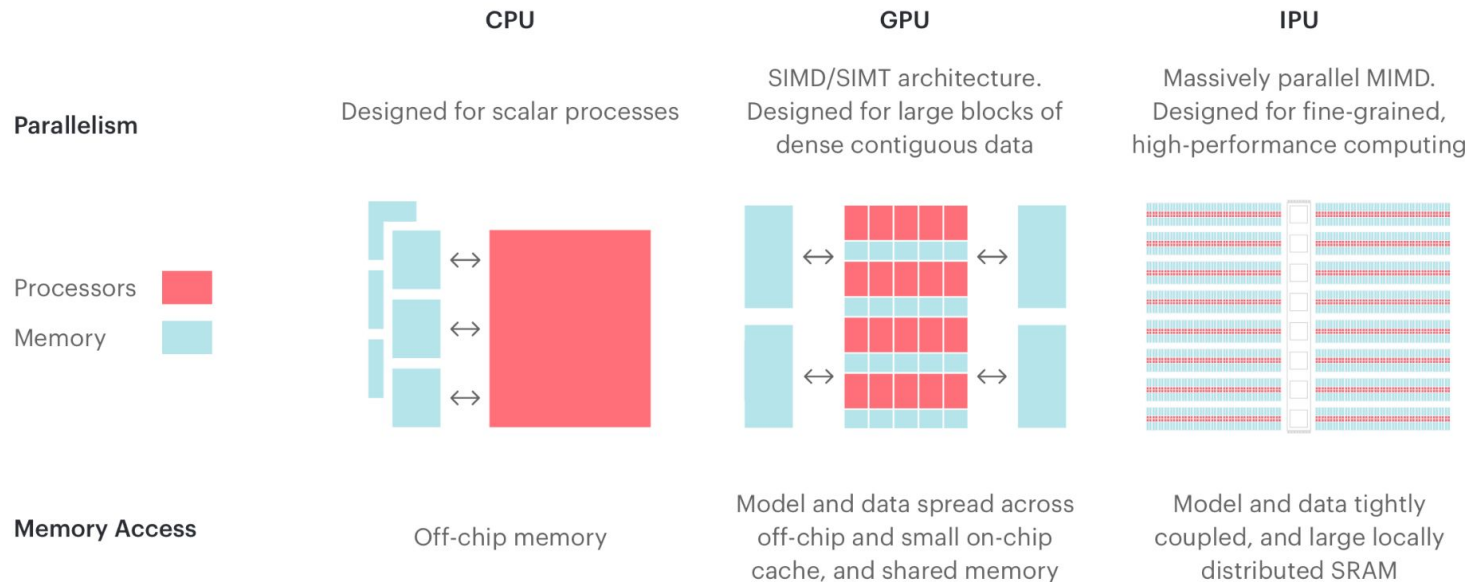
```
#SBATCH --gres=gpu:pvc:<number>
```

See our command-line tool to check configuration:

```
$ show_pvc_features
```

HOSTNAME	AVAIL_FEATURES	GRES	STATE
ac010	gen4_fabric	gpu:pvc:4	mixed
...			
ac024	gen4_fabric	gpu:pvc:8	idle
ac025	gen4_fabric	gpu:pvc:4	mixed
ac026	gen5_fabric	gpu:pvc:6	reserved
ac030	gen5_fabric	gpu:pvc:8	reserved
ac081	gen5_fabric,xelink4	gpu:pvc:4	reserved
ac082	gen5_fabric,xelink2	gpu:pvc:2	reserved
...			

Graphcore IPU



www.graphcore.ai/bow-processors

Graphcore IPU

GRAPHCORE

Porting TensorFlow 2 Models Quick Start

Version: Latest

 Search this document

- 1. Import the TensorFlow IPU module
- 2. IPU Config
- 3. Model
- 4. Training process
- 5. Optimization
- 6. Trademarks & copyright

Version: latest

 Download PDF

I. IMPORT THE TENSORFLOW IPU MODULE

First, we import the TensorFlow IPU module.

Add the import statement in [Listing 1.1](#) to the beginning of your script.

Listing 1.1 Importing ipu Python module

```
from tensorflow.python import ipu
```

For the `ipu` module to function properly, we must import it directly rather than accessing it through the top-level TensorFlow module.

2. IPU CONFIG

To use the IPU, you must create an IPU session configuration in the main process. A minimum configuration is in [Listing 2.1](#).

Listing 2.1 Example of a minimum configuration

```
ipu_config = ipu.config.IPUConfig()
ipu_config.auto_select_ipus = 1 # Select 1 IPU for the model
ipu_config.configure_ipu_system()
```

This is all we need to get a small model up and running. A full list of options is available in the [Python API documentation](#).

docs.graphcore.ai/en/latest

Accessing Graphcore IPU

- SSH into poplar1 or poplar2 from ACES

- `[username@login ~]$ ssh poplar1` ←

Contact us first to be given access to poplar
help@hprc.tamu.edu

- Enable the SDK environment. See our KB for details:

- hprc.tamu.edu/kb/User-Guides/ACES/Graphcore_Colossus_IPUs/
 - hprc.tamu.edu/kb/User-Guides/ACES/Graphcore_Bow_IPUs/

- Type **gc-monitor** to view the status of the IPU:

```
mouse@poplar1:~$ gc-monitor
```

gc-monitor Partition: p16 [active] has 16 reconfigurable IPU									
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	

IPU training event
([Link](#)) currently
planned for March 18th!

See also our Youtube
channel for previous
IPU training sessions!

System Software Stack

Function	Component	Version
Cluster Management	xCAT	2.16.4
Primary OS	Red Hat Enterprise Linux	8.9
HPC Scheduler	Slurm	22.05.11
InfiniBand Subnet Manager	UFM	6.12
OFED	Mellanox OFED	23.10-2.1.3
Storage Client	Lustre	2.12.9_ddn38
Software Management	Lmod	8.7
Software Build Framework	EasyBuild	4.9.2
Web Portal Software	Open OnDemand	3.0
Data Movement Software	Globus Connect Server	5.4
Job Reporting Software	Open XDMoD	10.5

Environment Setup

Lmod Module System

2,400+ Software Modules!

SOFTWARE MODULES ON THE ACES CLUSTER

ACES Software Modules

FASTER Software Modules

Grace Software Modules

Over 740 unique software packages – most with multiple version modules

(Last Updated: Thu Aug 15 14:22:54 CDT)

Name	Versions available	Description	Notes
ACTC	ACTC/1.1	'ACTC converts independent triangles into triangle strips or fans.'	
ANTLR	ANTLR/2.7.7-Java-11	'ANTLR, ANOther Tool for Language Recognition, (formerly PCCTS) is a language tool that provides a framework for constructing recognizers, compilers, and translators from grammatical descriptions containing Java, C#, C++, or Python actions.'	
AOCC	AOCC/4.0.0	'AMD Optimized C/C++ & Fortran compilers (AOCC) based on LLVM 13.0'	
APR	APR/1.7.0	'Apache Portable Runtime (APR) libraries.'	
APR-UTIL	APR-util/1.6.1	'Apache Portable Runtime (APR) util libraries.'	
ASE	ASE/3.20.1 ASE/3.22.1	'ASE is a python package providing an open source Atomic Simulation Environment in the Python scripting language. From version 3.20.1 we also include the ase-ext package, it contains optional reimplementations in C of functions in ASE. ASE uses it automatically when installed.'	
ATK	ATK/2.36.0 ATK/2.38.0	'ATK provides the set of accessibility interfaces that are implemented by other toolkits and applications. Using the ATK interfaces, accessibility tools have full access to view and control applications.'	

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

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

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/12.3.0 OpenMPI/4.1.5

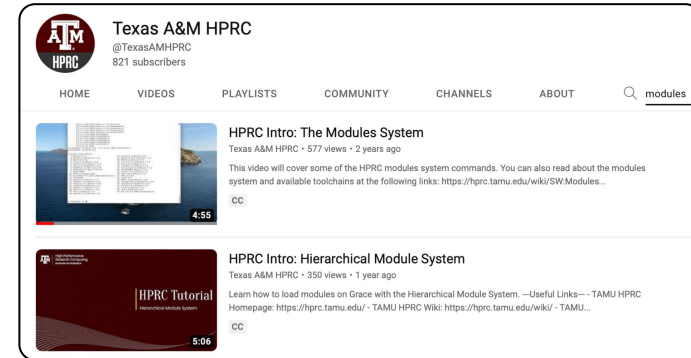
software-name / version-spec

 └─┬─┘ └─┬─┘ └─┬─┘

 PyTorch/2.1.2-CUDA-12.1.1

- 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 (will be named `module.avail.aces` or similar)

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 module 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 ~]$
```

Hands-on Exercise: Module Loading

1. `m1a pytorch` -See which versions of PyTorch are available.
2. `m1 PyTorch/2.1.2-CUDA-12.1.1` -error! You cannot do that yet.
3. `module spider PyTorch/2.1.2-CUDA-12.1.1` -Learn how to load this module.
4. `m1            PyTorch/2.1.2-CUDA-12.1.1` -Fill in the blank (with the correct toolchains) to load this module.
5. `m1 list` -List all loaded modules.
6. `m1 GCC/10.2.0` -Change version of a loaded Toolchain module (GCC); notice the message about reloaded modules.
7. `m1 list` -List all loaded modules.
8. `m1 purge` -Remove all loaded modules.

Python Virtual Environment

Python

- A Python virtual environment (venv) is an isolated environment where you can install packages without affecting the system-wide Python installation.

```
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  
Python/3.12.3
```

Create a Python Virtual Environment

```
# clean up and load Python
cd $SCRATCH
module purge
module load intelpython/2024.1.0_814

# create a Python virtual environment
python -m venv my-intel-ai-python-env

# activate the virtual environment
source my-intel-ai-python-env/bin/activate

# install required packages
pip install intel-extension-for-tensorflow[xpu]==2.15.0.1
intel-optimization-for-horovod==0.28.1.5
torch==2.1.0.post3 torchvision==0.16.0.post3
torchaudio==2.1.0.post3
intel_extension_for_pytorch==2.1.40+xpu
oneccl-bind-pt==2.1.400 deepspeed==0.14.2 numpy==1.26.4
--extra-index-url
https://pytorch-extension.intel.com/release-whl-aitools/

# deactivate the virtual environment
# deactivate
```

Conda Management System

Conda

Conda is an open-source package manager and environment management system that helps users install, manage, and deploy software packages, especially for Python, R, and other languages.

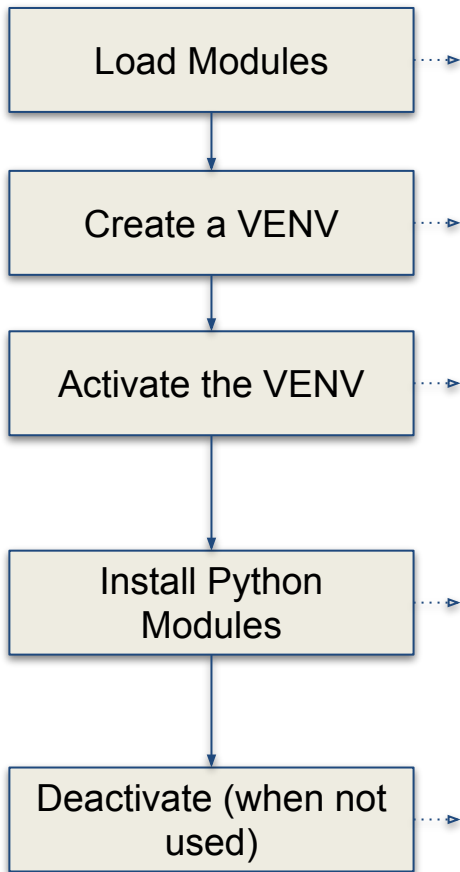
- Miniconda (lightweight)
- Anaconda (full distribution with many pre-installed packages)

Versions:

```
Miniconda3/4.12.0  
Miniconda3/22.11.1-1  
Miniconda3/23.3.1-0  
Miniconda3/23.5.2-0  
Miniconda3/23.9.0-0  
Miniconda3/23.10.0-1
```

Versions:

```
Anaconda3/2021.11  
Anaconda3/2022.05  
Anaconda3/2022.10  
Anaconda3/2023.03-1  
Anaconda3/2023.07-2  
Anaconda3/2024.02-1
```



```
# clean up and load Anaconda
cd $SCRATCH
module purge
module load Anaconda3/2024.02-1

# create a Python virtual environment
conda create -n ai-labs

# activate the virtual environment
source activate ai-labs

# install required package to be used in the portal
conda install -c anaconda jupyter
conda install -c anaconda pandas
conda install -c conda-forge matplotlib
conda install -c anaconda scikit-learn
conda install -c conda-forge transformers
...

# deactivate the virtual environment
# source deactivate
```

Containers

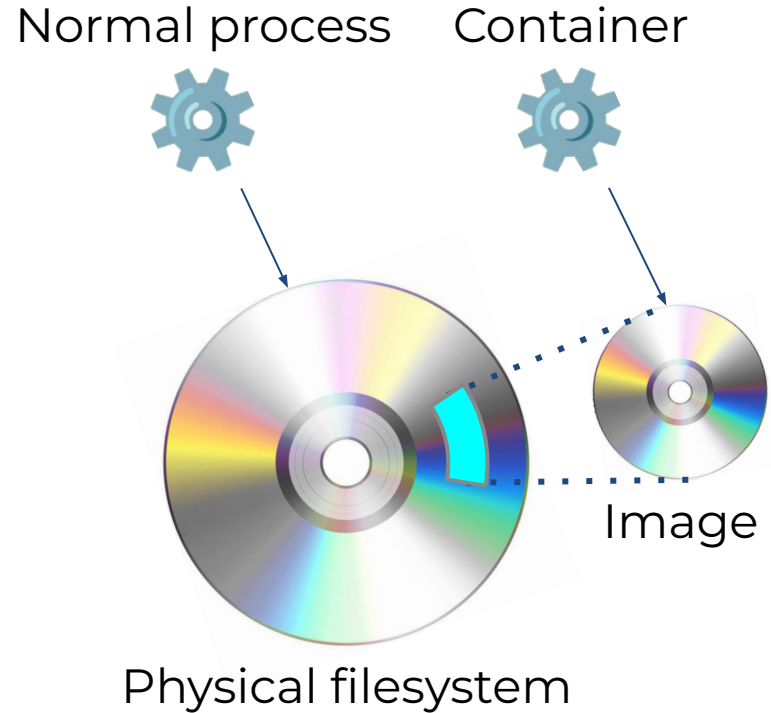
ACES: Fundamentals of Containers

ACES: Containers for Scientific Workflows (Singularity / Apptainer)

ACES: Containers for Scientific Workflows (Charliecloud)

What Are Containers?

- A container is a process (⚙️) that has its own **view** of local resources:
 - Filesystem
 - User IDs
 - Network
 - etc.
- Example: this container (⚙️ on the right) sees the **image** instead of the physical filesystem

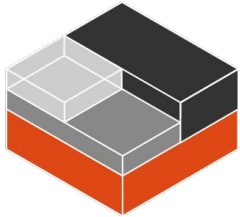


Why Use Containers?

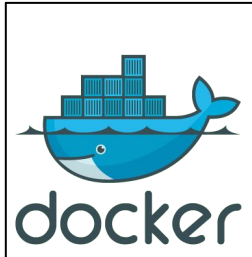
- **Shareability:**
 - Share your container image file by uploading to a public repository
 - Use images shared by others
- **Portability:**
 - Use images on any computer with the same architecture (x86-64)
- **Reproducibility:**
 - Container users are largely unaffected by changes to the cluster environments

Popular Container Runtimes

Instant deployment to users on different devices!



LXC
2008



Docker
2013

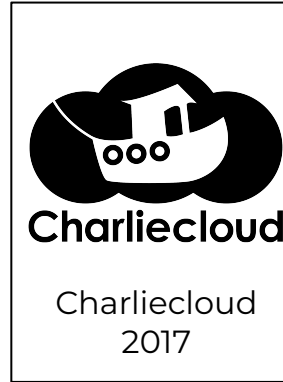


Singularity
2015



SHIFTER

Shifter 2016



Charliecloud
2017



Podman
2018

NVIDIA Container Registry Example

<https://catalog.ngc.nvidia.com/orgs/nvidia/containers/pytorch>

The screenshot shows the NVIDIA NGC Catalog interface for the PyTorch container. The top navigation bar includes the NVIDIA logo, 'NGC Catalog', a help icon, and 'Welcome Guest'. The left sidebar contains navigation icons for home, containers, and a specific container. The main content area is titled 'Containers > PyTorch' and features a 'Get Container' button. Below the title, there are tabs for 'Overview', 'Tags', 'Layers', 'Security Scanning', and 'Related Collections'. The 'Overview' tab is active, displaying the PyTorch logo, the text 'Accelerated with NVIDIA', and a description: 'PyTorch is an optimized tensor library for deep learning using GPUs and CPUs. Automatic differentiation is done with a tape-based system at both a functional and neural network layer level. This functionality brings a high level of flexibility and'. A 'Features' section at the bottom left indicates 'NVIDIA AI Enterprise Supported'.

Containers > PyTorch

PyTorch

Get Container

Overview Tags Layers Security Scanning Related Collections

PyTorch

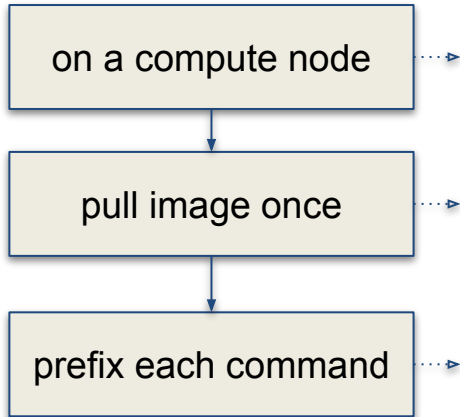
Accelerated with NVIDIA

PyTorch is an optimized tensor library for deep learning using GPUs and CPUs. Automatic differentiation is done with a tape-based system at both a functional and neural network layer level. This functionality brings a high level of flexibility and

Features

NVIDIA AI Enterprise Supported

NVIDIA Container Use Example



```
# set up Singularity workspace
cd $SCRATCH
module purge
module load WebProxy
export SINGULARITY_CACHEDIR=$TMPDIR

# fetch a container image (all on one line!)
singularity pull pytorch_25.sif \
docker://nvcv.io/nvidia/pytorch:25.01-py3

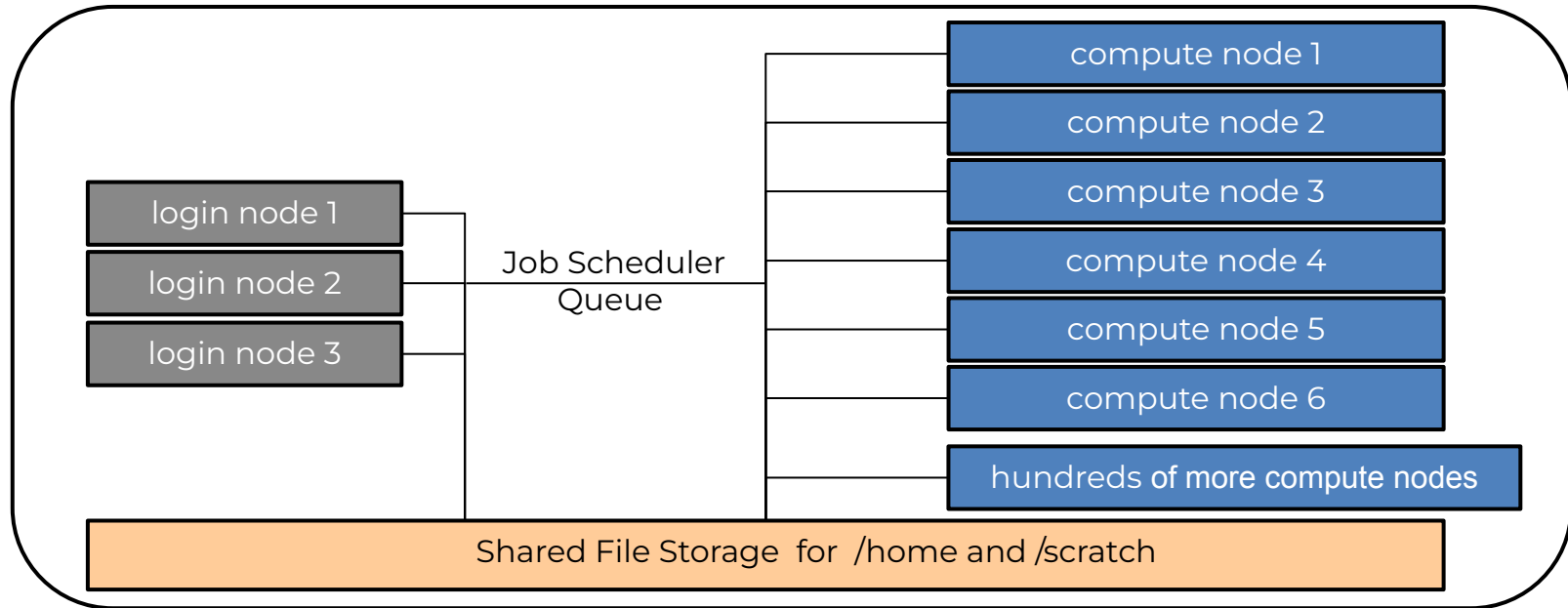
# execute your python scripts
singularity exec pytorch_25.sif python script.py
```

hprc.tamu.edu/kb/Software/Singularity/Examples/#nvidia-gpu-cloud

Software

ACES: Using the Slurm Scheduler on Composable Resources
ACES: Introduction to Composable Computing on ACES and FASTER

HPC Diagram and Slurm



login nodes are for:

- file manipulation and job script preparation
- software installation and testing
- short tasks (< 60 minutes and max 8 cores)
 - also be aware of amount of memory utilized

compute nodes are for:

- computational jobs which can use up to 96 cores and/or up to 488 GB memory per ACES compute node.
- all jobs running > 60 minutes

Slurm Job Script Example

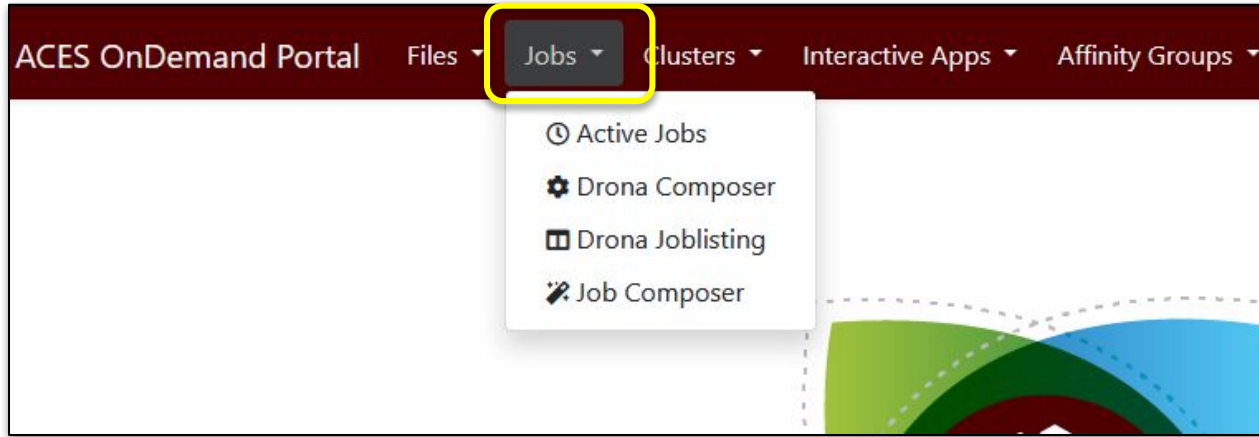
```
#!/bin/bash
#SBATCH --job-name=my_gpu_job
#SBATCH --time=1-00:00:00           # request 1 day of time for the job
#SBATCH --ntasks-per-node=1        # request 1 task (command)
#SBATCH --cpus-per-task=48         # request 1/2 of the available cores since using 1 of 2 GPUs
#SBATCH --mem=244G                 # request 1/2 of the available memory since using 1 of 2 GPUs
#SBATCH --gres=gpu:h100:1          # request 1 x H100 GPU; replace :1 with :2 for two GPUs
#SBATCH --partition=gpu            # use partition=gpu when selecting GPUs
#SBATCH --output=stdout.%x.%j
#SBATCH --error=stderr.%x.%j

# unload modules to start with a clean environment; then load required modules
module purge
module load CUDA/11.7.0
# run your gpu command
my_gpu_command
```

- There are two types of GPUs on ACES compute nodes. Select the type and quantity with `--gres`
 - H100 `--gres=gpu:h100:N` (N can be 1 to **max**) 30 x H100 on ACES
 - A30 `--gres=gpu:a30:N` (N can be 1 to **max**) 4 x A30 on ACES

The **max** value for **N** can change since ACES is a composable cluster

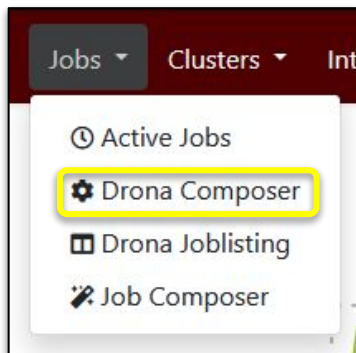
Portal Job Composers



Manage jobs without using the command line (will talk more about these later)

- “Active Jobs” is like `squeue`.
 - “Job Composer” lets you:
 - Create job scripts
 - Save job templates
 - Monitor your own jobs
- We will focus on the new “Drona” options:
- “Drona Joblisting” shows current and past jobs.
 - “Drona Composer” helps you build Slurm files via a form.

Portal: Drona Composer GUI



Import additional environments

Select customizable environments or workflows

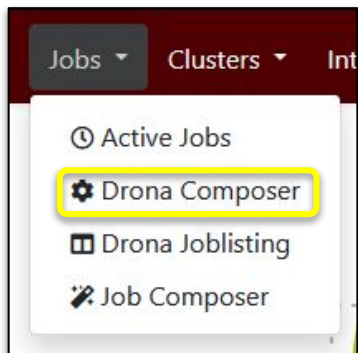
History of previously submitted jobs

Rerun jobs

A screenshot of the Drona Composer (ACES) Job Composer interface. The interface has a dark red header with the text 'DRONA COMPOSER (ACES)'. Below the header, there is a 'Job Composer' section. This section includes a 'Job Name' input field, a 'Location' input field with a 'Change' button, and an 'Environments' dropdown menu with a '+' button. Below these fields, there is a 'History' button. A table below the 'History' button shows a list of jobs. The table has columns for 'ID', 'Name', 'Location', 'Environment', 'Date', and 'Actions'. The table contains two rows of data. The first row has ID 632470, Name TestR, Location [drona_composer/runs/TestR](#), Environment R, Date 2025-01-15 15:27:59, and an Actions button. The second row has ID 632469, Name TestGeneric, Location [drona_composer/runs/TestGen...](#), Environment Generic, Date 2025-01-15 15:27:00, and an Actions button. At the bottom of the table, there is a 'Rows per page' dropdown set to 10, and a '1-2 of 2' indicator. There are also navigation arrows at the bottom right of the table.

ID	Name	Location	Environment	Date	Actions
632470	TestR	drona_composer/runs/TestR	R	2025-01-15 15:27:59	Actions
632469	TestGeneric	drona_composer/runs/TestGen...	Generic	2025-01-15 15:27:00	Actions

Portal: Drona Composer GUI



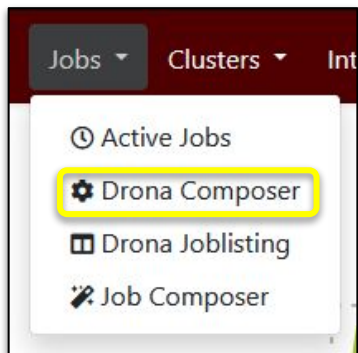
A screenshot of the Drona Composer GUI Job Composer form. The form is titled 'Job Composer' and is part of the 'DRONA COMPOSER (ACES)' interface. It contains several fields and sections for configuring a job. The 'Job Name' field is set to 'GenericRun'. The 'Location' field is set to '/scratch/user/u.mp108705/drona_composer/runs/GenericRun' and has a 'Change' button. The 'Environments' field is set to 'Generic' and has a dropdown arrow and a '+' button. The 'Upload files' field has a 'Select an option' dropdown and an 'Add' button. The 'Add modules' field has a 'Default (foss/2023b)' dropdown and an 'Add' button with a 'PyTorch2.12-CUDA-12.1.1' module selected. The 'Number of tasks' field is set to '100'. The 'Advanced task options' section has a checked checkbox. The 'Number of nodes' field is set to '1'. The 'Number of cpus per tasks' field is empty. The 'Use Accelerator' field is set to 'H100'. The '#GPUs' field is set to '1'. The 'TOTAL Memory' field is empty with a 'GB' unit selector. The 'Expected run time' field is set to '3' days. The 'Project Account' field is set to '155062417651 (910.12)'. The 'Additional Slurm parameters' field is empty. There are 'Preview' and 'Hide History' buttons at the bottom right.

Generic environment to create custom Slurm batch job

Add modules for job

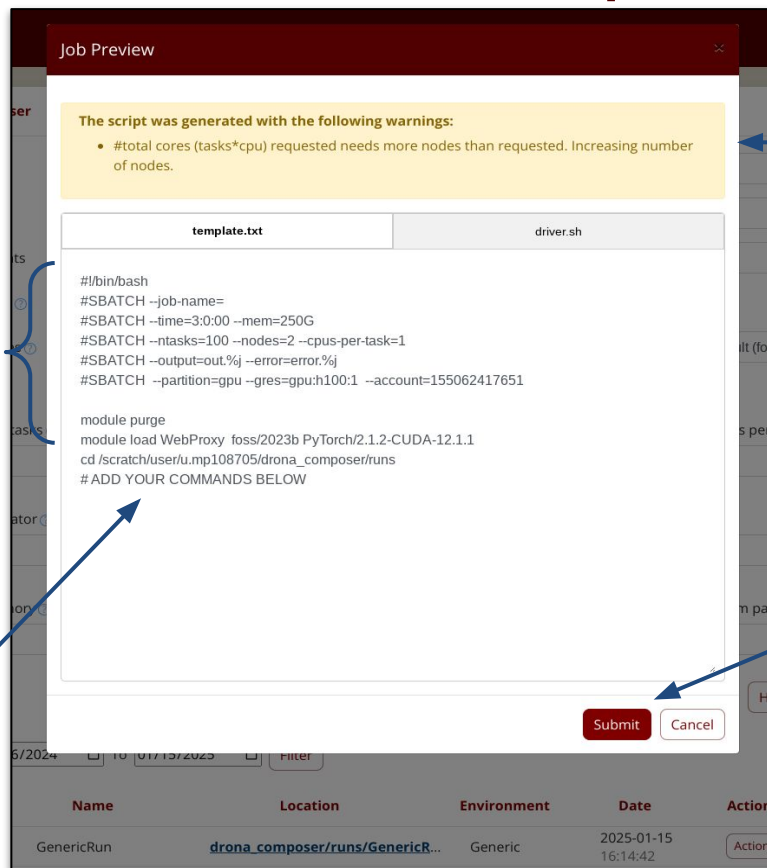
Fill out info that Slurm would need

Portal: Drona Composer GUI



Generated Slurm directives and module loads

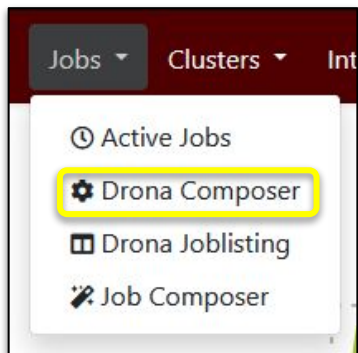
Add commands directly into editable window



Pane with Input specific warnings

Hit "Submit"

Portal: Drona Composer GUI



PyTorch and
TensorFlow
Containers

A screenshot of the Drona Composer (ACES) Job Composer form. The form is titled 'Job Composer' and is located in the center of the interface. It has a dark red header with the text 'ATM High Performance Research Computing DIVISION OF RESEARCH' and 'DRONA COMPOSER (ACES)'. The form contains several input fields: 'Job Name' (a text box), 'Location' (a text box with a 'Change' button), and 'Environments' (a dropdown menu). The 'Environments' dropdown menu is open, showing a list of options: 'Generic', 'Matlab', 'R', 'Python', 'intel-tf-container' (highlighted), 'nvidia-pt-container', 'intel-pt-container', and 'nvidia-tf-container'. A blue arrow points from the text 'PyTorch and TensorFlow Containers' to the 'intel-tf-container' option in the dropdown menu.

Popular Distribution Strategies

- Data Parallel (DP)
- Distributed Data Parallel (DDP)
- Fully Sharded Data Parallel (FSDP)
- Tensor Parallelism
- Pipeline Parallelism
- ZeRO (Zero Redundancy Optimizer)
- MirroredStrategy
- MultiWorkerMirroredStrategy
- ...

Intel® oneAPI Toolkits

Intel® oneAPI Base Toolkit

A core set of high-performance libraries and tools for building C++, SYCL and Python applications



Add-on Domain-specific Toolkits



Intel® oneAPI Tools for HPC

Deliver fast Fortran, OpenMP
& MPI applications that scale



Intel® oneAPI Tools for IoT

Build efficient, reliable solutions
that run at network's edge



Intel® oneAPI Rendering Toolkit

Create performant, high-fidelity
visualization applications

Toolkits powered by oneAPI



Intel® AI Analytics Toolkit

Accelerate machine learning & data science
pipelines end-to-end with optimized DL
frameworks & high-performing Python
libraries



Intel® Distribution of OpenVINO™ Toolkit

Deploy high performance inference &
applications from edge to cloud

(Source: Intel)

Intel® AI Tools Selector



Select a tool to start [?](#)

AI Tools OpenVINO™ Toolkit Intel® Gaudi® Software

Conda®-based Binary Installer Package - All Tools [?](#)

Offline Installer

Choose a preset OR customize [?](#)

Data Analytics Classical Machine Learning Deep Learning

Inference Optimization **Customize**

Distribution Type [?](#)

conda* **pip** Docker*

Python® Versions [?](#)

Python® 3.9 Python 3.10

PyTorch® Framework Optimizations [?](#)

☐ Intel® Extension for PyTorch® (CPU)

☒ Intel® Extension for PyTorch® (GPU)

① All packages are for Linux® only. For compatibility details, refer to the [System Requirements](#).

AI Tools: Customize

Customize your tool selections for conda and pip. Docker containers are not available for customizations.

Set Up Your Environment

Install with pip

```
pip install intel-extension-for-tensorflow[xpu]==2.15.0.1 intel-optimization-for-horovod==0.28.1.5 torch==2.1.0.post3 torchvision==0.16.0.post3 torchaudio==2.1.0.post3 intel_extension_for_pytorch==2.1.40+xpu onecccl-bind-pt==2.1.400 deepspeed==0.14.2 numpy==1.26.4 --extra-index-url https://pytorch-extension.intel.com/release-whl-aitools/
```

Verify Installation

Run a Get Started Sample

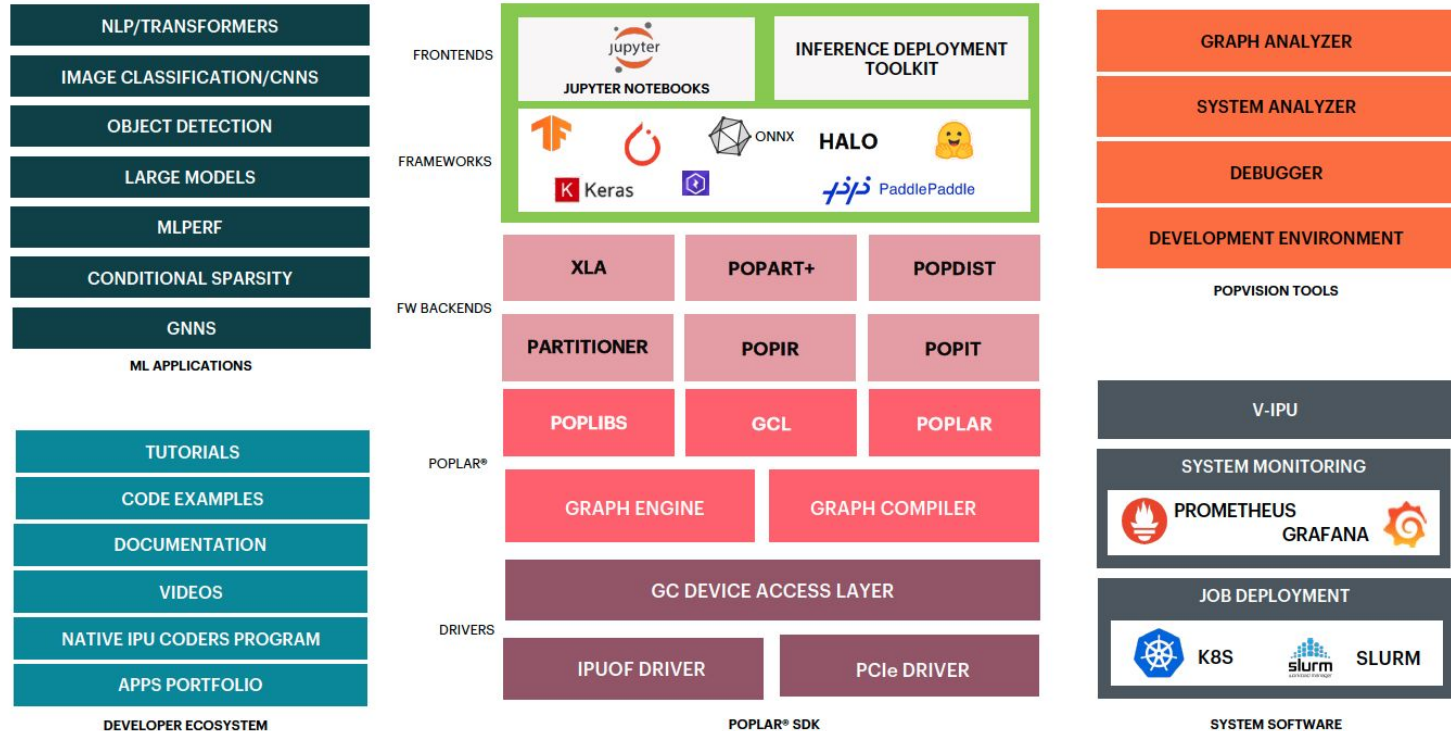
Next Steps

[Intel® AI Reference Models](#) (formerly Model Zoo) repository contains links to pretrained models, sample scripts, best practices, and tutorials for many popular open source machine learning models optimized by Intel.

[Working with Docker Containers](#) document provides more information about preset containers and instructions on how to

(<https://www.intel.com/content/www/us/en/developer/tools/oneapi/ai-tools-selector.html>)

Graphcore Software Stack



Poplar SDK setup

```
source  
/opt/gc/poplar/poplar_sdk-ubuntu_20_04-3.4.0+1507-69d9d03fd8/poplar-ub  
untu_20_04-3.4.0+73-aa67dd6164/enable.sh
```

```
source  
/opt/gc/poplar/poplar_sdk-ubuntu_20_04-3.4.0+1507-69d9d03fd8/popart-ub  
untu_20_04-3.4.0+73-aa67dd6164/enable.sh
```

```
mkdir -p /localdata/$USER/tmp  
export TF_POPLAR_FLAGS=--executable_cache_path=/localdata/$USER/tmp  
export POPTORCH_CACHE_DIR=/localdata/$USER/tmp  
export TORCH_HOME=/localdata/$USER/tmp/
```

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

Hugging Face on



```
[u.zh108696@aces-login3 ~]$ module load GCC/13.2.0 huggingface_hub/0.27.1
[u.zh108696@aces-login3 ~]$ huggingface-cli login

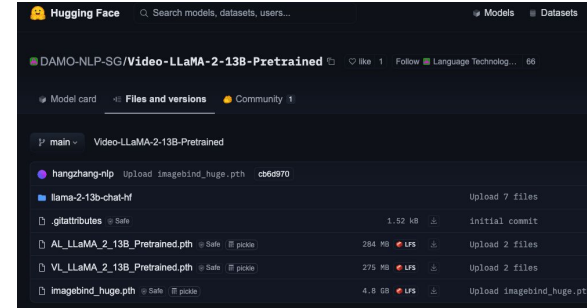
Enter your token (input will not be visible):

A token is already saved on your machine. Run 'huggingface-cli whoami' to get more information or 'huggingface-cli logout' if you want to log out.

Setting a new token will erase the existing one.
To log in, 'huggingface_hub' requires a token generated from https://huggingface.co/settings/tokens .
Enter your token (input will not be visible):
```

Hugging Face Hub 🧡 huggingface_hub

- huggingface_hub (models, datasets, etc)
 - git-lfs (Git Large File Storage)
- Using Git LFS (Large File Storage) with the Hugging Face Hub is crucial for handling large files like machine learning models and datasets.



Hugging Face Libraries 📖 LIBRARIES

- Transformers
- Datasets
- Tokenizers
- Accelerate
- Diffusers
- ...

```
(hgf-env) [u.zh108696@aces-login3 huggingface]$ pip install transformers datasets tokenizers
Looking in indexes: https://pypi.org/simple, https://pypi.ngc.nvidia.com
Collecting transformers
  Downloading transformers-4.48.0-py3-none-any.whl.metadata (44 kB)
    44.4/44.4 kB 5.7 MB/s eta 0:00:00
Collecting datasets
  Downloading datasets-3.2.0-py3-none-any.whl.metadata (20 kB)
Collecting tokenizers
```

```
[u.zh108696@aces-login3 ~]$ ml spider git-lfs

git-lfs:

Description:
  Git Large File Storage (LFS) replaces large files
  while storing the file contents on a remote server

Versions:
  git-lfs/3.2.0
  git-lfs/3.3.0
  git-lfs/3.5.1
```

```
from transformers import AutoTokenizer, AutoModelForSequenceClassification

# Choose a model, e.g., BERT for sentiment analysis
model_name = "distilbert-base-uncased-finetuned-sst-2-english"

# Load tokenizer and model
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForSequenceClassification.from_pretrained(model_name)

# Tokenize input
inputs = tokenizer("I love ACES!", return_tensors="pt")

# Perform inference
outputs = model(**inputs)
predictions = outputs.logits.argmax(dim=-1)
print(predictions) # 1 for positive sentiment
```



Performance Monitor and Optimization

System Management Interface SMI

- **nvidia-smi**
- **xpumcli**
- **sysmon**
- **gc-monitor**

hprc.tamu.edu/kb/User-Guides/ACES/Intel_PVCs/#sysmon

hprc.tamu.edu/kb/User-Guides/ACES/Graphcore_Bow_IPUs/#gc-monitor

```
Every 2.0s: nvidia-smi
```

ac064: Sun Feb 16 15:11:54 2025

Sun Feb 16 15:11:54 2025

NVIDIA-SMI 535.216.01				Driver Version: 535.216.01		CUDA Version: 12.2	
GPU	Name	Perf	Persistence-M	Bus-Id	Disp.A	Volatile	Uncorr. ECC
Fan	Temp		Pwr:Usage/Cap		Memory-Usage	GPU-Util	Compute M.
							MIG M.
0	NVIDIA A30	A30	On	00000000:34:00:0	Off	69%	0
N/A	20C	P0	71W / 165W	549MiB / 24576MiB			Default Disabled
1	NVIDIA A30	A30	On	00000000:35:00:0	Off	56%	0
N/A	20C	P0	75W / 165W	549MiB / 24576MiB			Default Disabled

Processes:							GPU Memory Usage
GPU	GI ID	CI ID	PID	Type	Process name		
0	N/A	N/A	1662124	C	...3/2024.02-1/envs/ai-labs/bin/python		540MiB
1	N/A	N/A	1662125	C	...3/2024.02-1/envs/ai-labs/bin/python		540MiB

num_workers=0

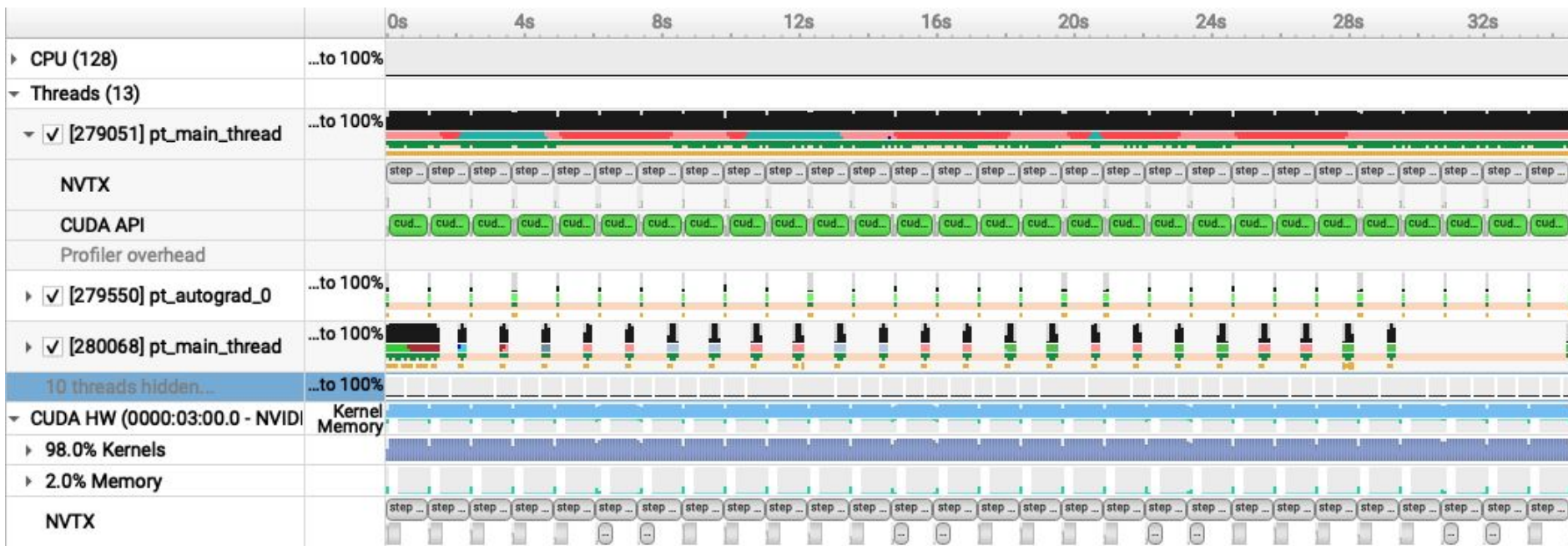
NVIDIA Nsight



<https://github.com/NERSC/sc24-dl-tutorial>

num_workers=4

NVIDIA Nsight



<https://github.com/NERSC/sc24-dl-tutorial>

TensorBoard

Visualization and debugging tool for machine learning experiments

Originally developed for **TensorFlow**, it now supports **PyTorch, JAX, and standalone logging**.

What Can TensorBoard Do?

It helps track and visualize various aspects of your ML training, including:

Scalars – Track metrics like loss, accuracy, learning rate over time.

Graphs – Visualize the model's computational graph.

Histograms – Analyze weight distributions and activations.

Images/Audio/Text – Inspect samples from datasets or model outputs.

Embeddings – Explore high-dimensional data using PCA or t-SNE.

Hyperparameter Tuning – Compare multiple training runs.

Profiling – Monitor GPU/CPU usage, memory, and bottlenecks.

TensorBoard on ACES portal

The screenshot shows the ACES OnDemand Portal interface. At the top is a dark red navigation bar with the following links: ACES OnDemand Portal, Files, Jobs, Clusters, Interactive Apps, Affinity Groups, Chatbot, and Dashboard. The main content area features the ACES logo (a stylized green and blue square with a white circuit-like pattern) and the text "ACES ACCELERATING COMPUTING FOR EMERGING SCIENCES". Below the logo, there is a section titled "Message of the Day" and a section titled "IMPORTANT POLICY INFORMATION" which contains a list of rules: "Unauthorized use of HPRC resources is prohibited", "Use of HPRC resources in violation of United States", "Sharing HPRC account and password information is DISABLED.", and "Authorized users must also adhere to ALL policies". A dropdown menu is open from the "Interactive Apps" link, showing a list of applications: GUI, VNC, NextSilicon VNC, Imaging, CryoSPARC, ImageJ, Jmol, Paraview, cisTEM, Servers, Jupyter Notebook, JupyterLab, RStudio, TensorBoard (highlighted with a red box and a red arrow), and Tutorials OnDemand. At the bottom of the page, there is a warning: "!! WARNING: THERE ARE ONLY NIGHTLY BACKUPS OF USER HOME DIRECTORIES. !!".

ACES OnDemand Portal Files Jobs Clusters Interactive Apps Affinity Groups Chatbot Dashboard

GUI

VNC

NextSilicon VNC

Imaging

CryoSPARC

ImageJ

Jmol

Paraview

cisTEM

Servers

Jupyter Notebook

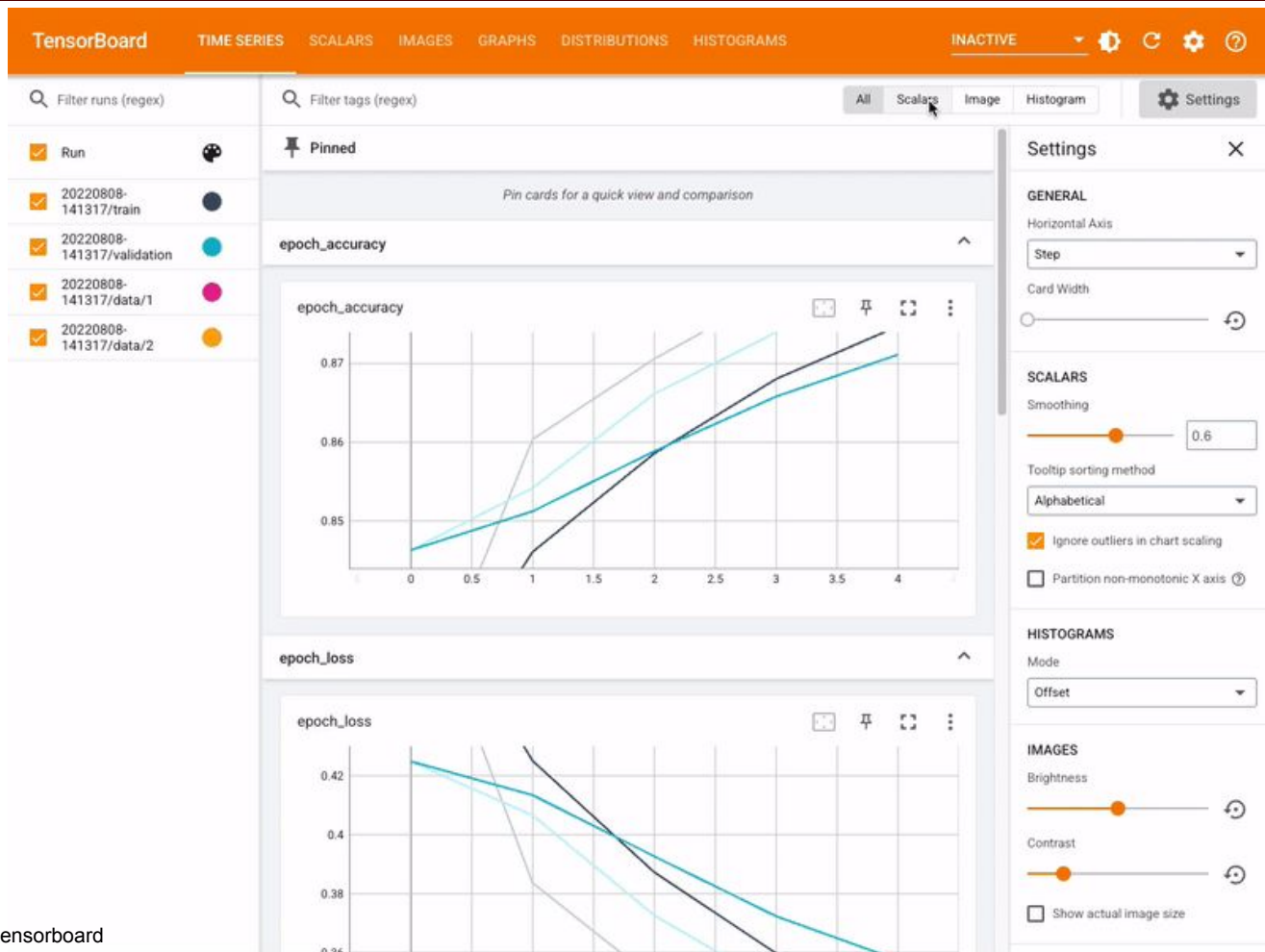
JupyterLab

RStudio

TensorBoard

Tutorials OnDemand

!! WARNING: THERE ARE ONLY NIGHTLY BACKUPS OF USER HOME DIRECTORIES. !!



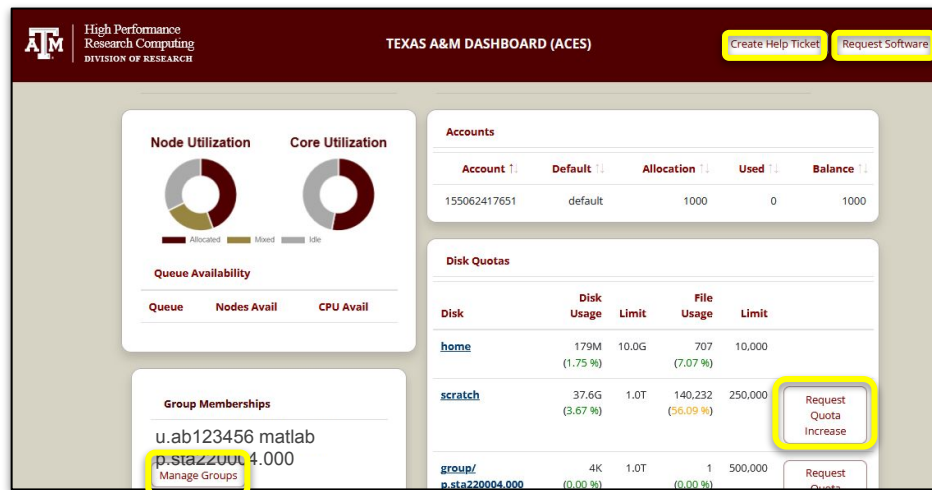
<https://www.tensorflow.org/tensorboard>

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**
TEXAS A&M UNIVERSITY

<https://hprc.tamu.edu>

HPRC Helpdesk:

help@hprc.tamu.edu

Phone: 979-845-0219

Take our short course survey!



HPRC Survey

https://u.tamu.edu/hprc_shortcourse_survey

Help us help you. Please include details in your request for support, such as, Cluster (ACES, FASTER, Grace, Launch), NetID (UserID), Job information (JobID(s), Location of your jobfile, input/output files, Application, Module(s) loaded, Error messages, etc), and Steps you have taken, so we can reproduce the problem.



Documentation and Training

HPRC Knowledge Base

The screenshot shows the HPRC Knowledge Base website. The header is dark red with the Texas A&M HPRC logo and a search bar. The main navigation bar includes links for Home, Quick Start, User Guides, Software, Helpful Pages, and FAQ. The 'User Guides' section is expanded, showing a list of topics including ACES, Hardware, Key Policies, Accessing, Computing Environment, File Systems, Compiling and Running Code, Batch System, Graphcore Colossus IPUs, Graphcore Bow IPUs, ACES Support for AI and ML, MemVerge, NextSilicon, Vector Engine, and Intel PVC GPUs. The 'Hardware' section is selected, displaying the title 'ACES: A Dell x86 HPC Cluster'. A table provides details about the system, including the system name, operating system, total compute cores/nodes, and a list of compute nodes. A 'Table of contents' sidebar on the right lists the sections: ACES: A Dell x86 HPC Cluster, Compute Nodes, System Interconnect, and Data Transfer Nodes.

ACES: A Dell x86 HPC Cluster	
System Name:	ACES
Operating System:	Red Hat Enterprise 8
Total Compute Cores/Nodes:	11,888 cores 130 nodes
Compute Nodes:	110 Intel Sapphire Rapids Nodes 17 Intel Ice Lake Nodes 1 AMD Rome Graphcore Node with 16 Mk2 Colossus GC200 IPUs 1 Intel Ice Lake Graphcore Node with 16 Bow-2000 IPUs 1 Intel Cascade Lake Node with 8 NEC Vector Engine Type 20B-P cards
Composable Components	30 NVIDIA H100 GPUs (In 15 SPR nodes, Composability Coming Soon)

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

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

Training on YouTube

High Performance Research Computing
DIVISION OF RESEARCH

Texas A&M HPRC
@TexasAMHPRC · 1.35K subscribers · 107 videos
The Texas A&M HPRC YouTube channel assists HPRC users with training on HPRC resources.
hprc.tamu.edu and 1 more link
Subscribe

<https://www.youtube.com/texasamhprc>

HOME VIDEOS **PLAYLISTS** COMMUNITY CHANNELS ABOUT

Created playlists

- Tutorial**
Jump-Host Tunnel
Windows Powershell
3 videos
ACES: Getting Started
View full playlist
- Short Course**
Intermediate Linux
10 videos
ACES
View full playlist
- Application for a TACC Batch Account**
2 videos
Project Account Management (AMS)
View full playlist
- Using the LSF Job Scheduler on Ada**
31 videos
Previous Primers and Short Courses:
View full playlist
- Interview with LEARN President and CEO, Akbar Kara**
8 videos
BRICCs
View full playlist
- HPRC OpenOnDemand Portal**
6 videos
HPRC OpenOnDemand Portal
View full playlist

Accelerator Training

IPU Labs

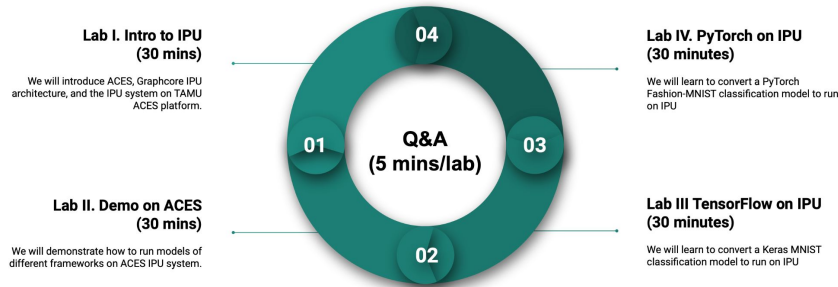
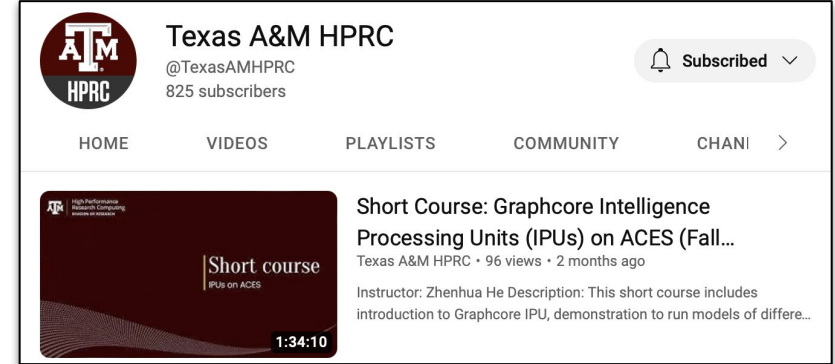


Figure 1. Structure of the IPU Training Laboratories.



This semester we will have training sessions for IPUUs and PVC GPUs:

- PVCs - Mar 4th - https://hprc.tamu.edu/training/aces_intel_pvc.html
- IPUUs - Mar 18th - https://hprc.tamu.edu/training/aces_ipus.html

See also our “ACES Training” YouTube Playlist for previous courses:

- <https://www.youtube.com/@TexasAMHPRC/playlists>

Getting Started on the ACES Cluster

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

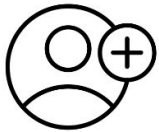
Allocations Management



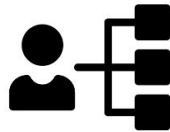
You can get allocations on both **ACES** and **FASTER** through ACCESS.



ACES can also be allocated through *NAIRR*.



CREATE
ACCOUNT



SELECT
OPPORTUNITY



REQUEST
ALLOCATION



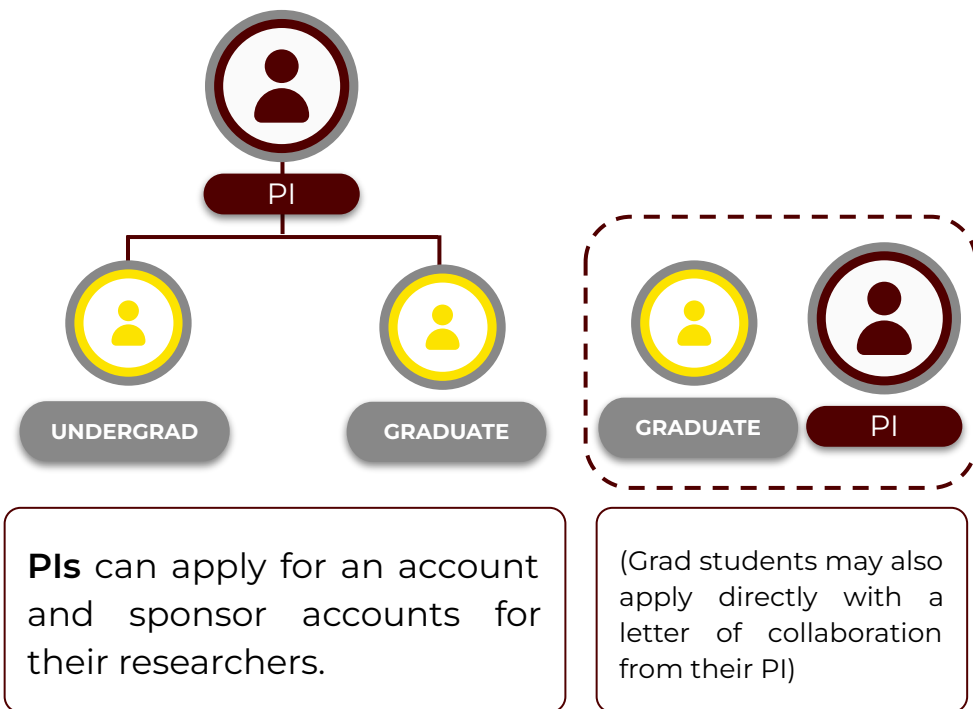
RECEIVE
CREDITS



EXCHANGE
CREDITS

Getting on ACES

- Using an [ACCESS](https://access-ci.org) 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.





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

ACES Dashboard

ACES OnDemand Portal

Files ▾

Jobs ▾

Clusters ▾

Interactive Apps ▾

Affinity Groups ▾

Dashboard ▾

My Interactive Sessions

ACES Dashboard



High Performance
Research Computing
DIVISION OF RESEARCH

TEXAS A&M DASHBOARD (ACES)

Create Help Ticket

Request Software

Node Utilization

Core Utilization



Allocated Mixed Idle

Queue Availability

Queue

Nodes Avail

CPU Avail

Group Memberships

u.ab123456 matlab p.sta220004.000

Manage Groups

Accounts

Account ↑↓	Default ↑↓	Allocation ↑↓	Used ↑↓	Balance ↑↓
155062417651	default	1000	0	1000

Disk Quotas

Disk	Disk Usage	Limit	File Usage	Limit
home	179M (1.75 %)	10.0G	707 (7.07 %)	10,000
scratch	37.6G (3.67 %)	1.0T	140,232 (56.09 %)	250,000
group/ p.sta220004.000	4K (0.00 %)	1.0T	1 (0.00 %)	500,000

Request
Quota
Increase

Request help or
new software

Request higher
storage limits

Request or change user groups

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



Utilities

The image shows the ACES OnDemand Portal interface. The top navigation bar includes 'Affinity Groups', 'Dashboard', and 'Utilities' (highlighted with a yellow box). The 'Utilities' dropdown menu lists: 'cpuavail', 'envsavail', 'gpuavail', 'license_status', and 'venvavail'. Below the navigation bar, two utility windows are displayed. The first window, 'ACES GPU Availability', shows the output of the 'gpuavail' command, including a table of GPU configuration and availability. The second window, 'ACES License Status', shows the output of the 'license_status -a' command, including a list of available licenses and a table of license status for Abaqus.

ACES GPU Availability

This displays the CONFIGURATION of installed GPUs and the status of GPUs readily available for jobs. GPUs and GPU nodes not in the AVAILABILITY table are busy with other jobs and will be available after the job is completed.

Output generated: 2025-01-21 08:26:16

```
$ gpuavail
```

CONFIGURATION		AVAILABILITY					
NODE TYPE	NODE COUNT	NODE NAME	GPU TYPE	GPU COUNT	GPU AVAIL	CPU AVAIL	GB MEM AVAIL
gpu.pvc:4	17	ac041	h100	8	8	96	488
gpu.pvc:2	11	ac045	h100	8	8	96	488
gpu.pvc:8	4	ac049	h100	4	3	88	360

ACES License Status

Output generated: 2025-01-21 08:26:57

```
$ license_status -a
```

Abaqus | ANSYS | COMSOL | ConvergeCFD | ECLIPSE | FDTD | FEMZIP | FLOW3D |
Hyperworks | Intel | INTERSECT | LS-DYNA | Madymo | Matlab | Matlab_MDCS |
Metashape | MOE | Osays | PGL_compilers | StarCCM | Schrodinger |

License status for Abaqus:

License Name	# Issued	# In Use	# Available
abaqus_teaching	320	0	320
abaqus_extended_teaching	40	0	40
cse_teaching	1	0	1
euler_lagrange_teaching	1	0	1

- convenient collection of command-line tools
- check availability of gpus, licenses, and other resources