ACES: Software for AI on HPC

February 18, 2025

Zhenhua He & Richard Lawrence

High Performance
Research Computing
DIVISION OF RESEARCH

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.

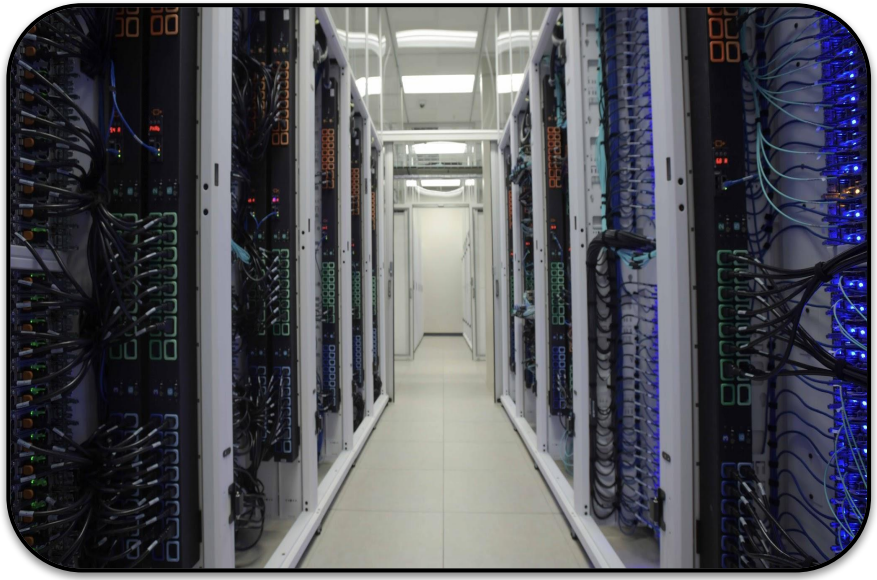
HPC provides the **necessary infrastructure** to:

- 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**

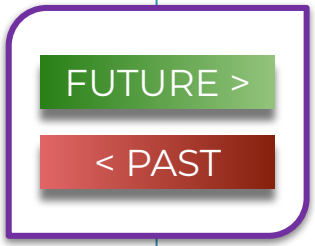


See hprc.tamu.edu/resources
for more info on all our clusters

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

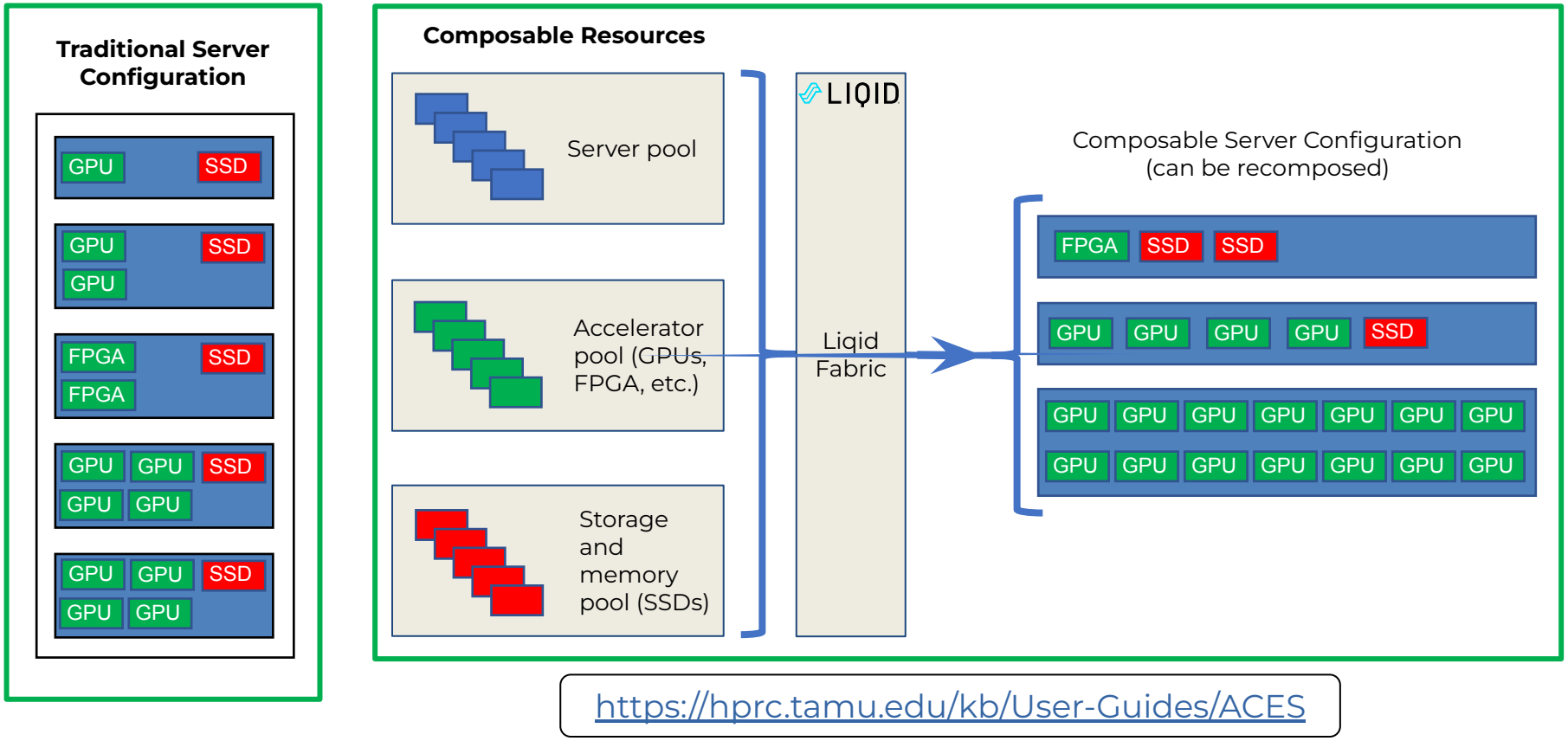


HPC for AI

- 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

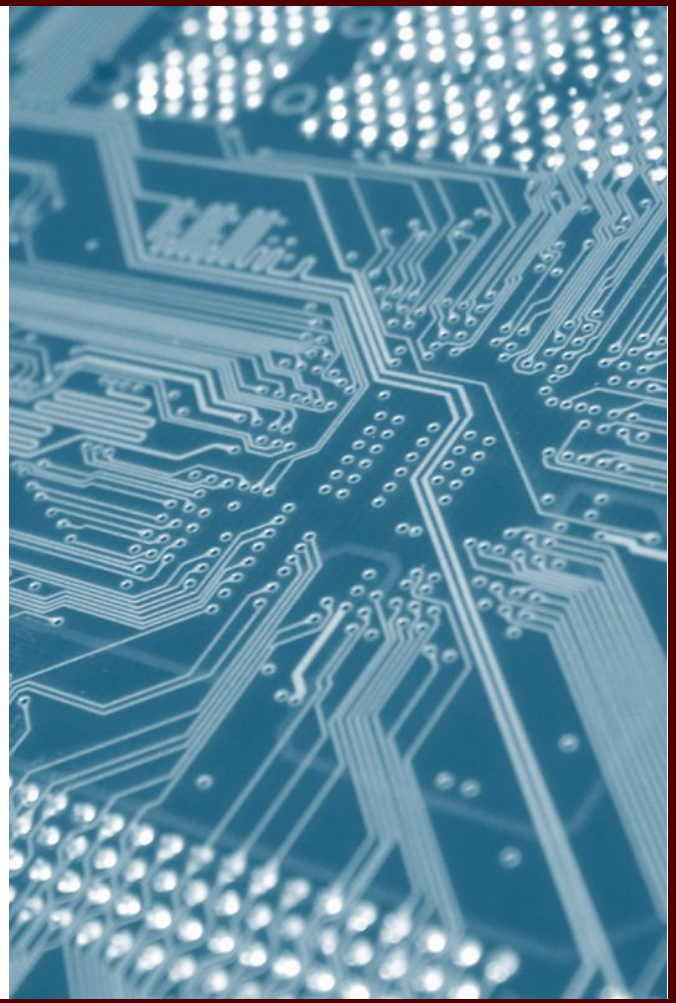


https://hprc.tamu.edu/kb/User-Guides/ACES

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 Liqid handle composable configurations automatically
- *Note*: The IPUs and NEC Vector Engines cannot be composed.

ACES Overview

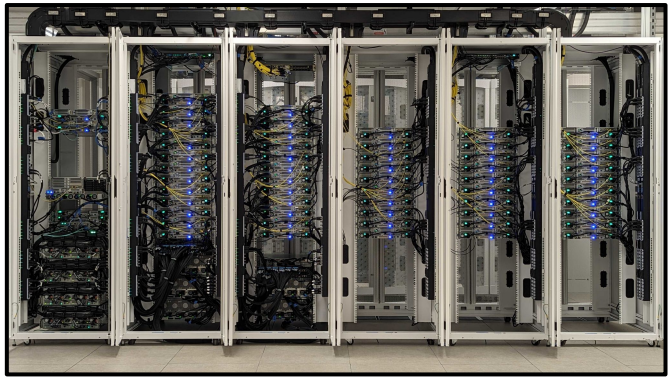
NSF ACES

Accelerating Computing for Emerging Sciences

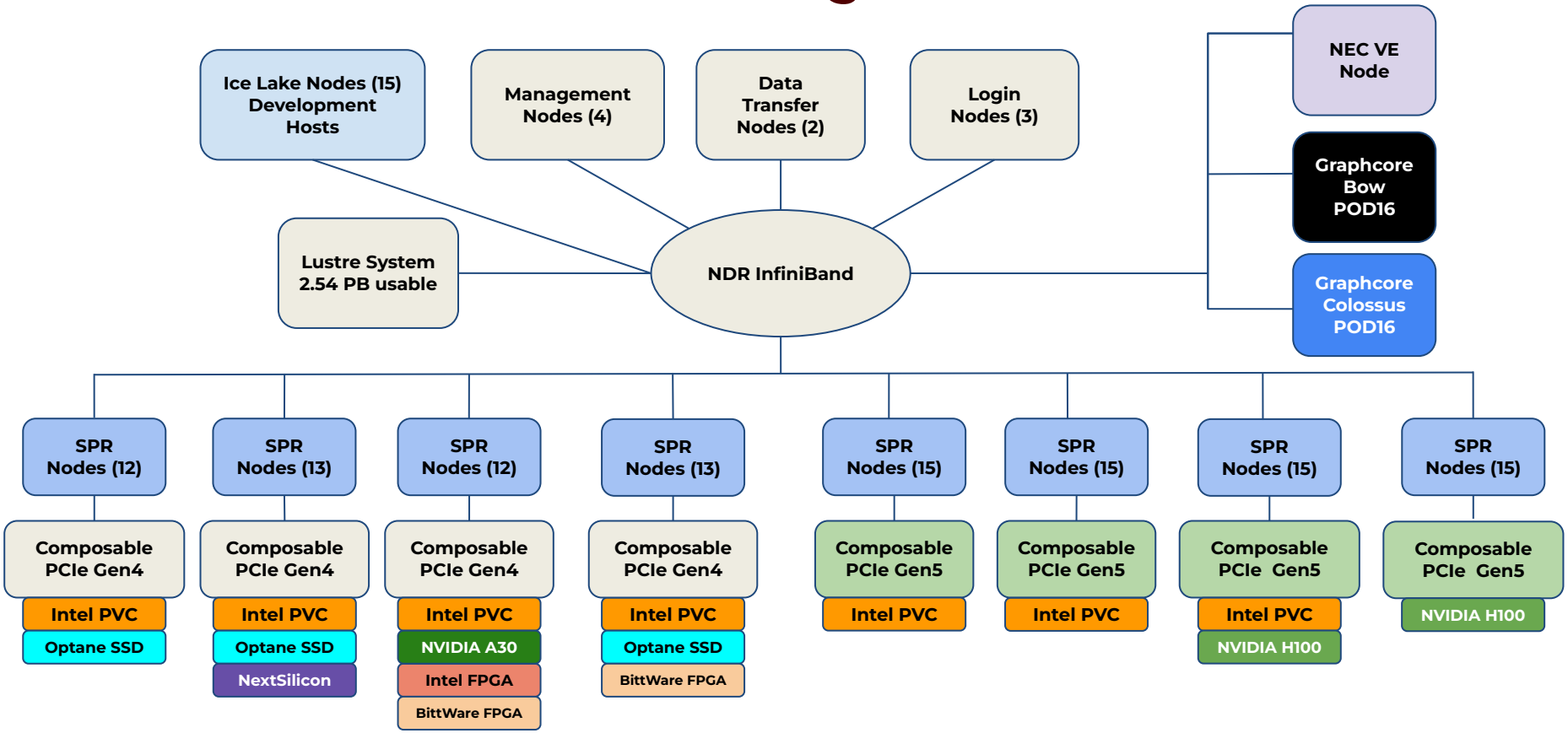
Our Mission:

- 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

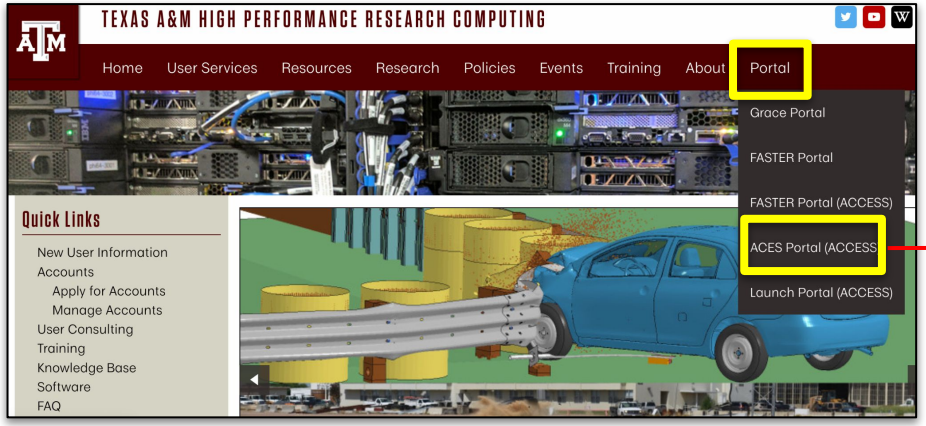
ACES Accelerators/Components

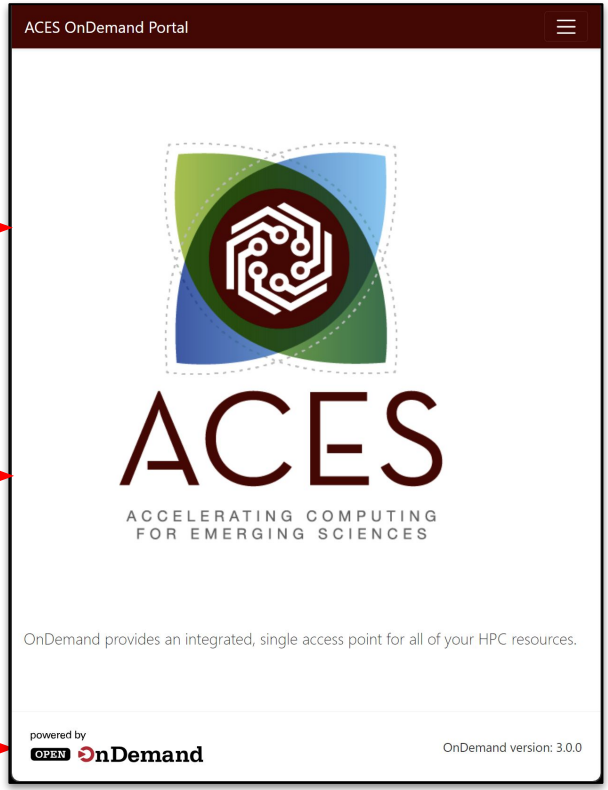
Component	Quantity	Description
Graphcore IPU	32	16 Colossus GC200 IPUs, 16 Bow IPUs. Each IPU group hosted with a CPU server as a POD16 on a 100 GbE RoCE fabric
FPGAs: Intel PAC D5005 BittWare IA-840F	 2 3	 Accelerator with Intel Stratix 10 GX FPGA and 32 GB DDR4 Accelerator with Agilex AGF027 FPGA and 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.
NVIDIA GPUs: H100 A30	 30 4	 For HPC, DL Training, AI Inference 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



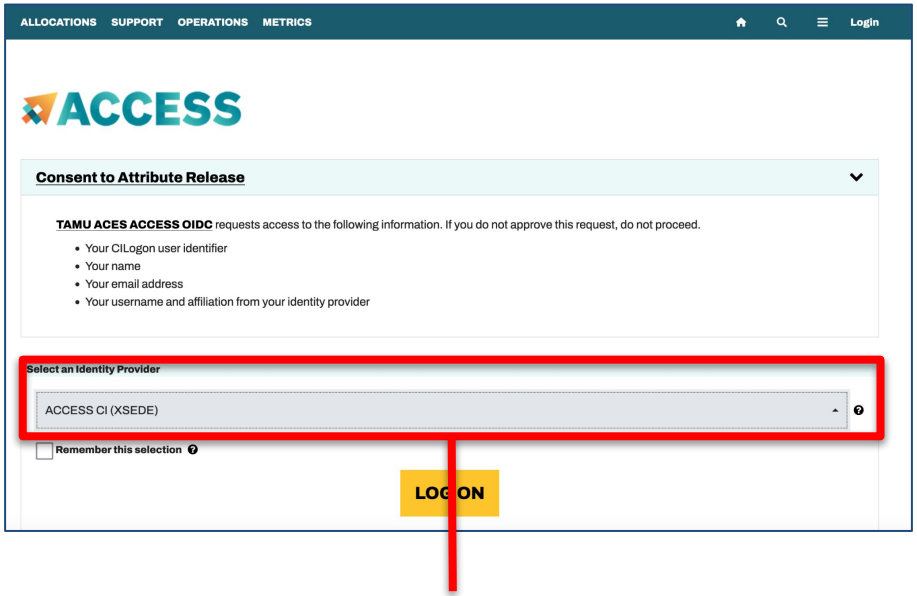


ACES Portal portal-aces.hprc.tamu.edu
is the web-based user interface for the ACES cluster

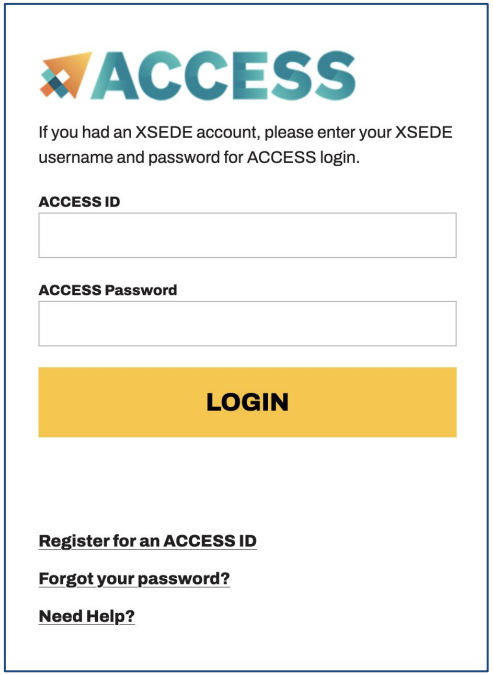
[HPRC Portal YouTube tutorials]

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

Accessing ACES via the Portal (ACCESS)



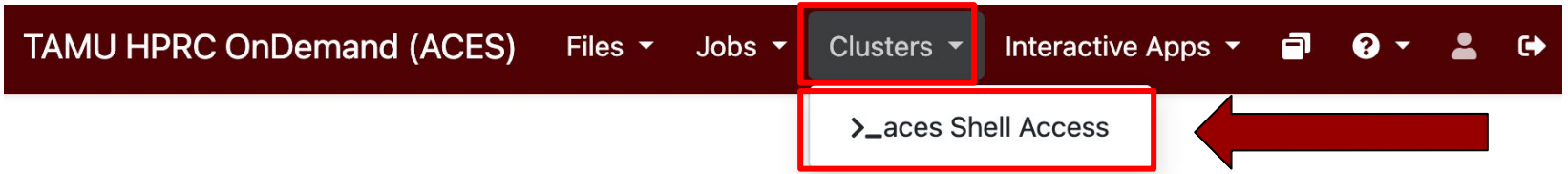
Select the Identity Provider appropriate for your account.



Log in using your ACCESS or institutional credentials.

Get a Shell on ACES

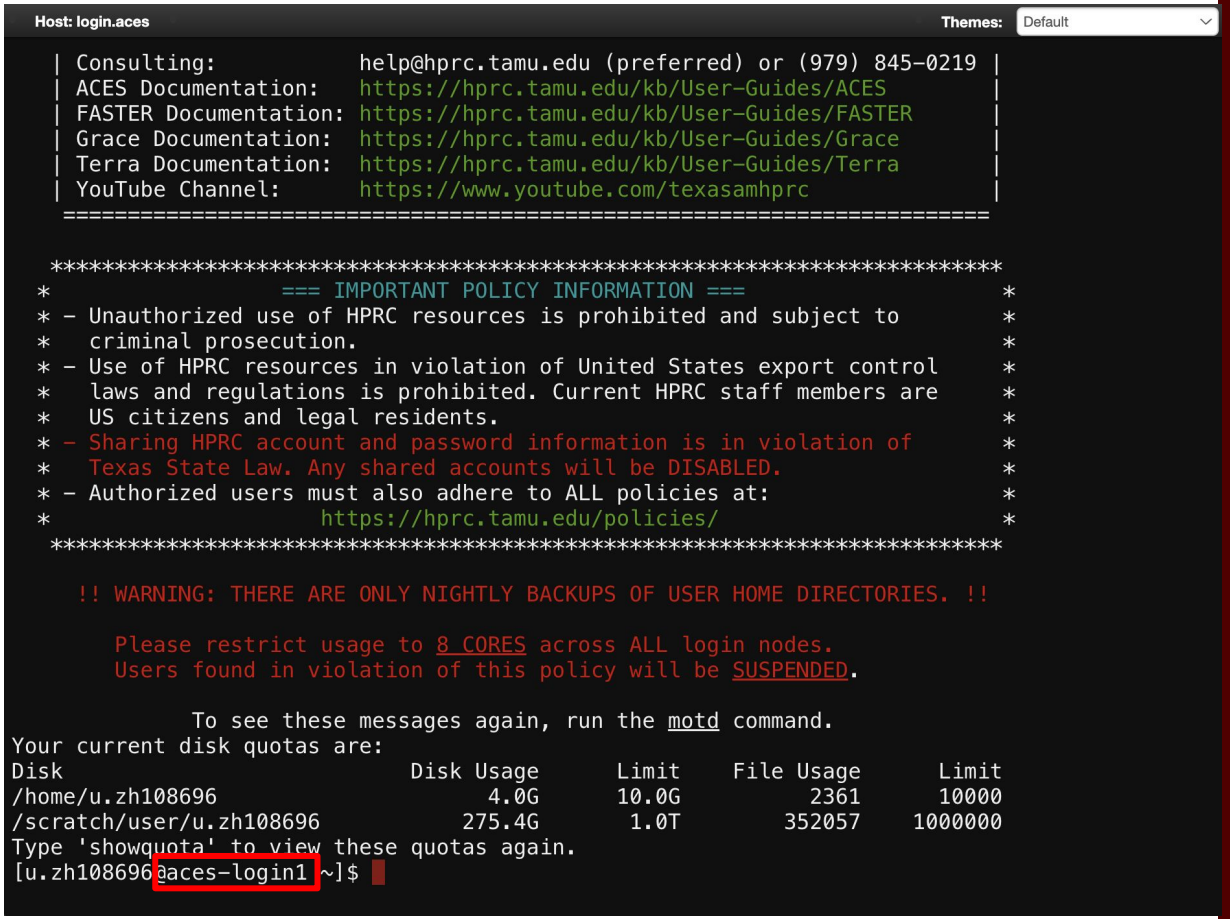
Click on “Clusters” menu → _aces Shell Access



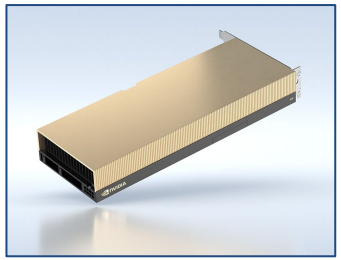
Success!

Welcome to the ACES login node.

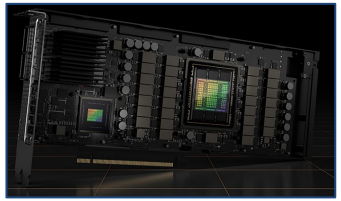
Check which login node you are on.



NVIDIA GPUS on ACES



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

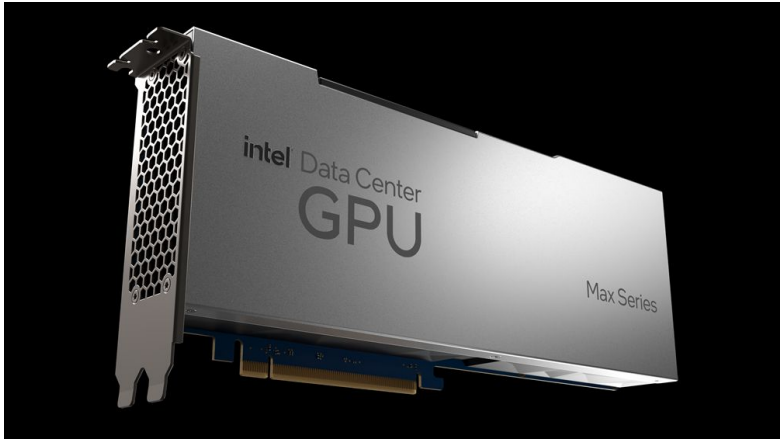


H100s Support computationally intensive workloads employing numerical simulations and AI/ML methods.

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

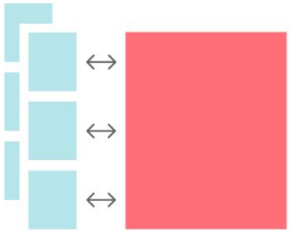
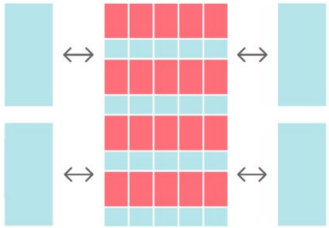
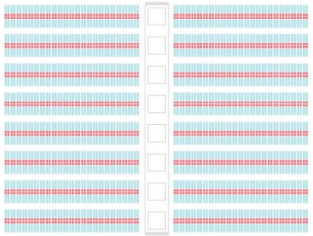
Specify in Slurm file:

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

See our command-line tool to check configuration:

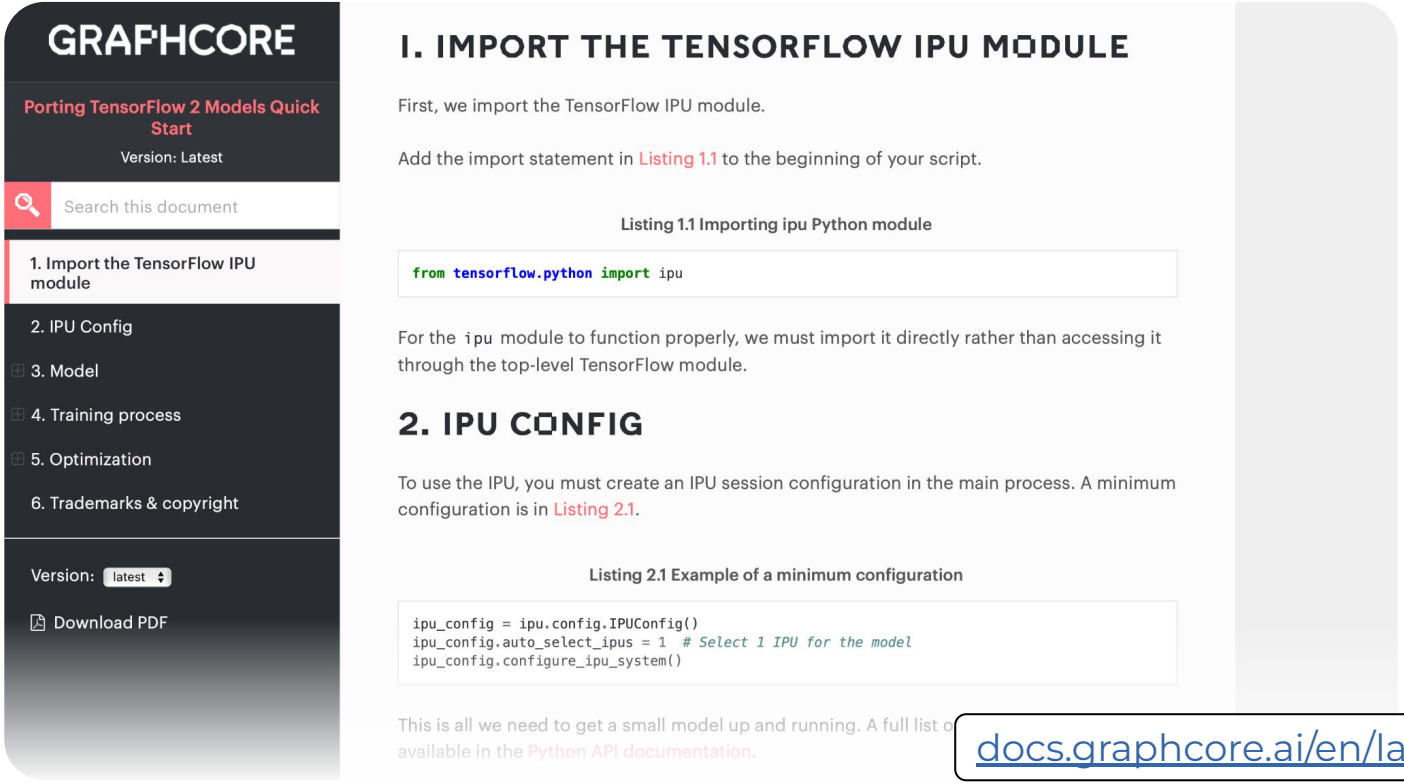
```
$ show_pvc_features
HOSTNAME AVAIL_FEATURES           GRES          STATE
ac010    gen4_fabric              gpu:pvc:4     mixed
...
ac024    gen4_fabric              gpu:pvc:8     idle
ac025    gen4_fabric              gpu:pvc:4     mixed
ac026    gen5_fabric              gpu:pvc:6     reserved
ac030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

Graphcore 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



docs.graphcore.ai/en/latest

Accessing Graphcore IPUs

- 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 IPUs:

```
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  |
```

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

(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	

Over 740 unique software packages – most with multiple version modules

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

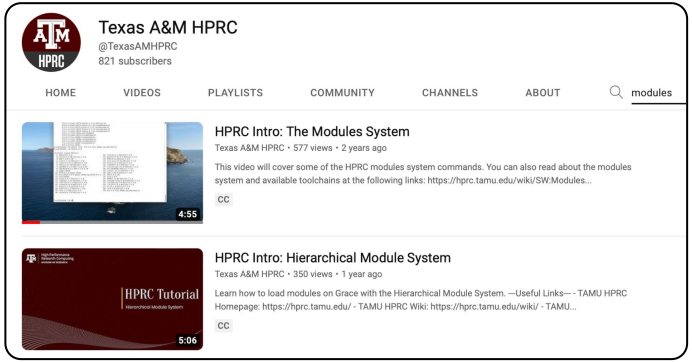
- Uses the command: **module**
- Each version of a software, application, library, etc. is available as a module.
  - Module names have the format:

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 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 ~]$
```

Hands-on Exercise: Module Loading

1. `mla pytorch` -See which versions of PyTorch are available.
2. `ml 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. `ml _____ PyTorch/2.1.2-CUDA-12.1.1` -Fill in the blank (with the correct toolchains) to load this module.
5. `ml list` -List all loaded modules.
6. `ml GCC/10.2.0` -Change version of a loaded Toolchain module (GCC); notice the message about reloaded modules.
7. `ml list` -List all loaded modules.
8. `ml 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

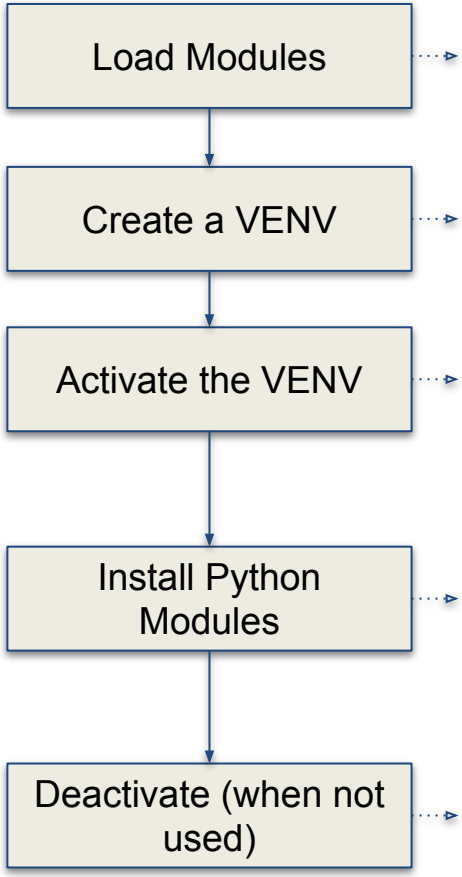
Conda

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

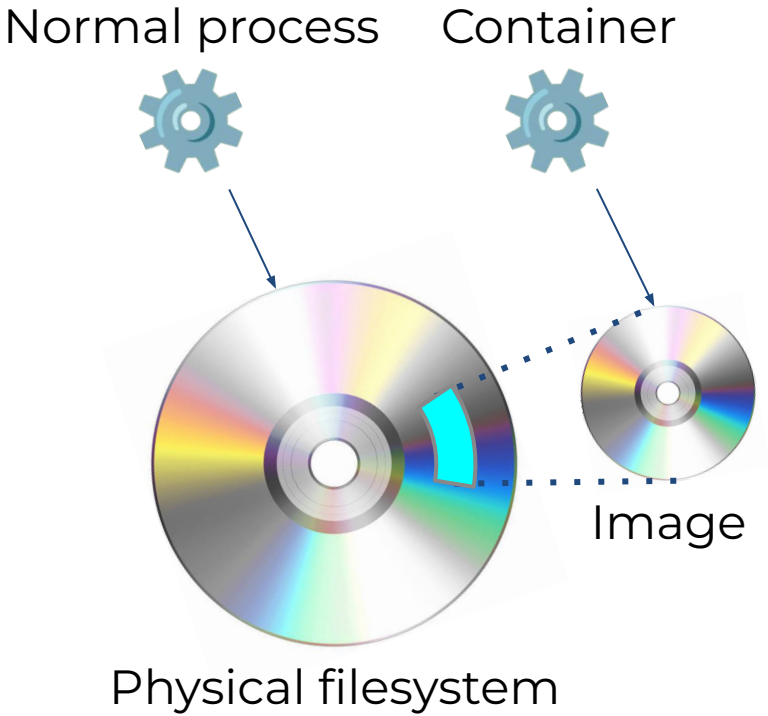
ACES: Containers for Scientific Workflows (Singularity / Apptainer)

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

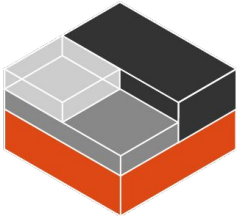


Why Use Containers?

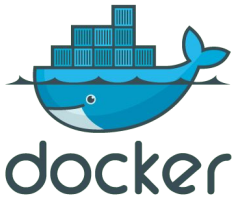
- **Shareability**:
  - Share your container image file by uploading to a public repository
  - Use images shared by others
- **Portability**:
  - Use images on any computer with the same architecture (x86-64)
- **Reproducibility**:
  - Container users are largely unaffected by changes to the cluster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

Popular Container Runtimes

Instant deployment to users on different devices!



LXC
2008



Docker
2013

Singularity
2015

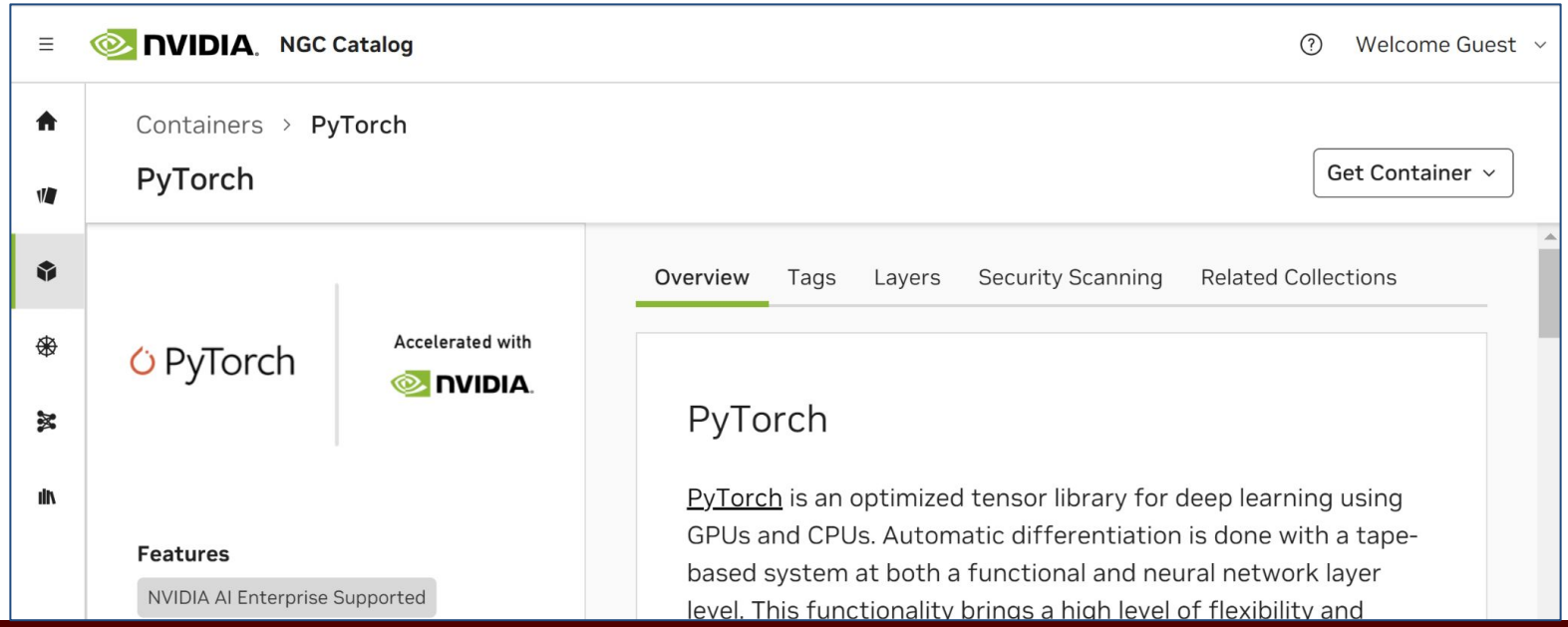
Shifter 2016

Charliecloud
2017

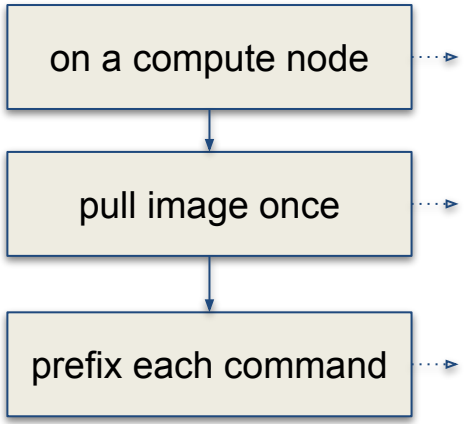
Podman
2018

NVIDIA Container Registry Example

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



NVIDIA Container Use Example



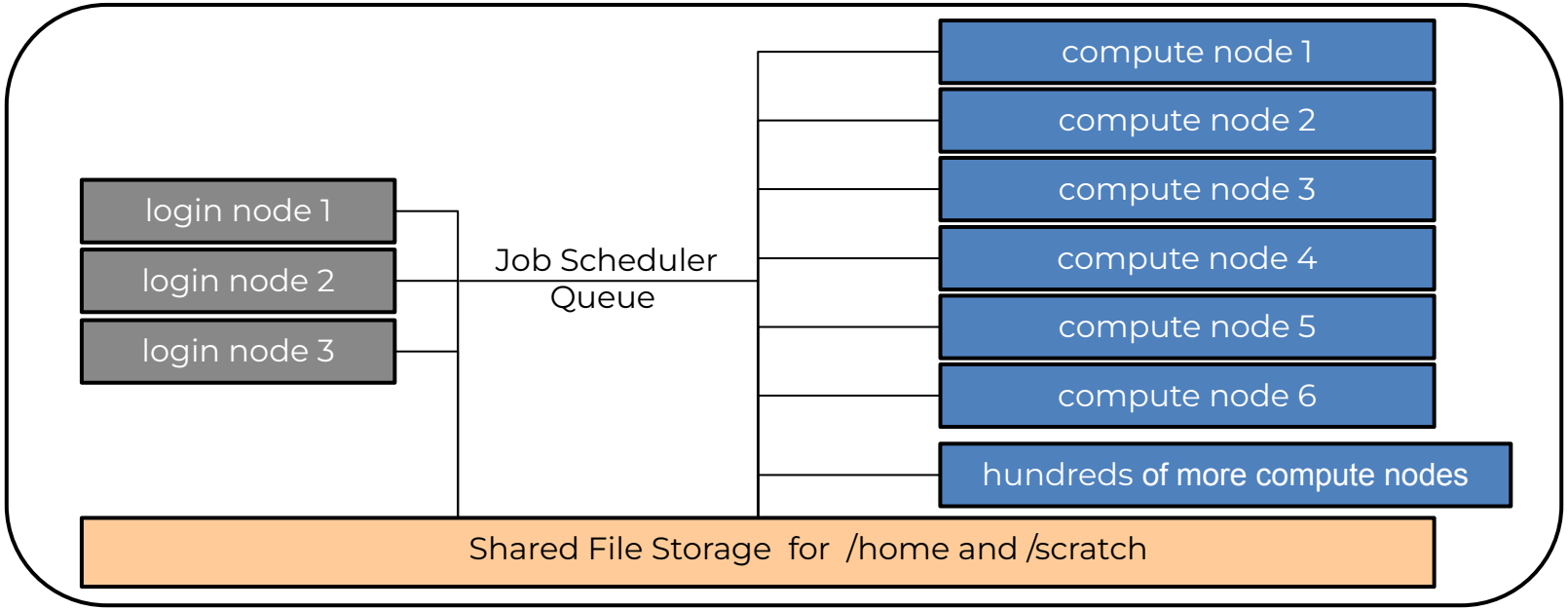

```
# set up Singularity workspace
cd $SCRATCH
module purge
module load WebProxy
export SINGULARITY_CACHEDIR=$TMPDIR

# fetch a container image (all on one line!)
singularity pull pytorch_25.sif \
docker://nvcr.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

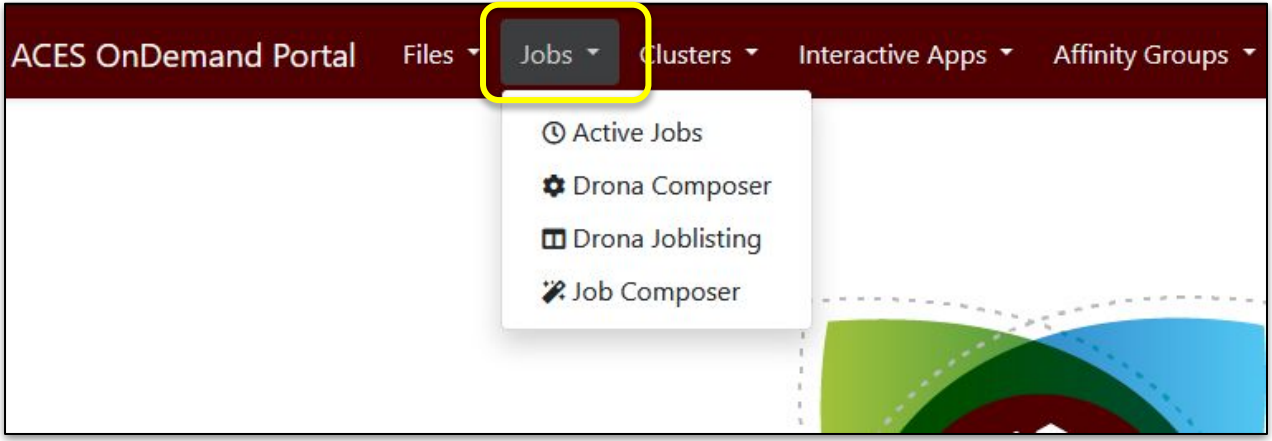
Slurm Job Script Example

```
#!/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 ½ of the available cores since using 1 of 2 GPUs
#SBATCH --mem=244G                 # request ½ 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



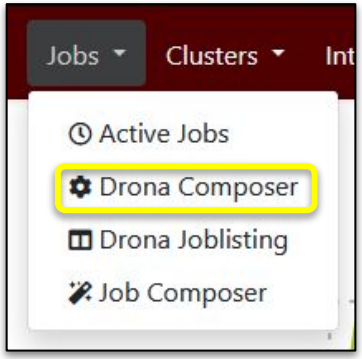
Manage jobs without using the command line (will talk more about these later)

- "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

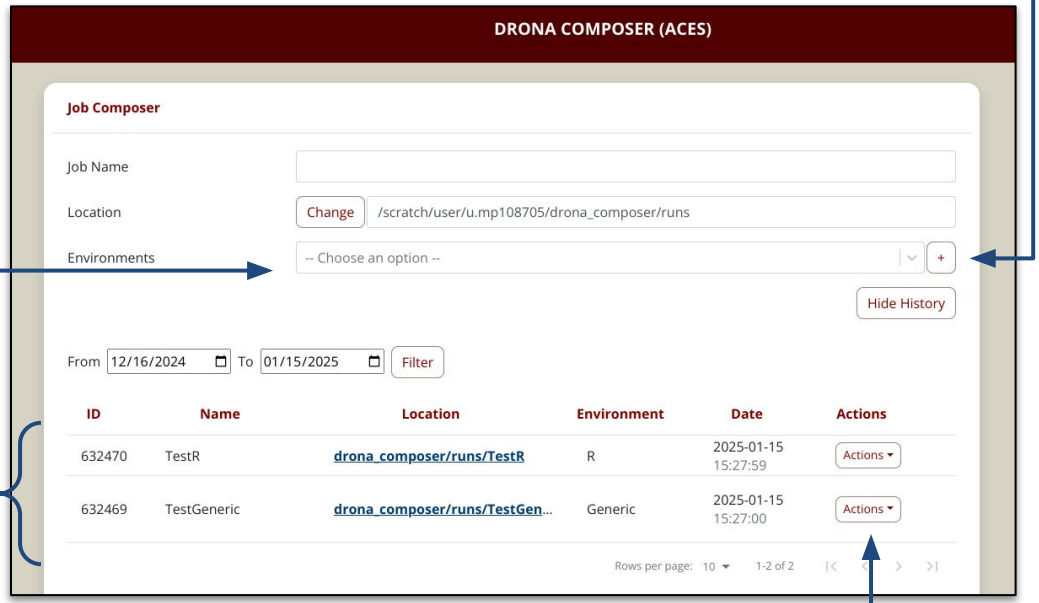


Import additional environments

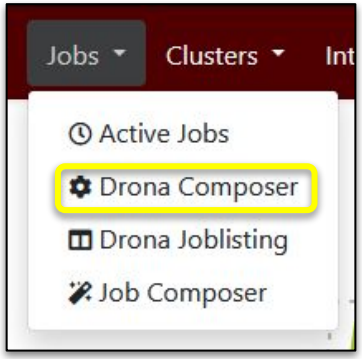
Select customizable environments or workflows

History of previously submitted jobs

Rerun jobs



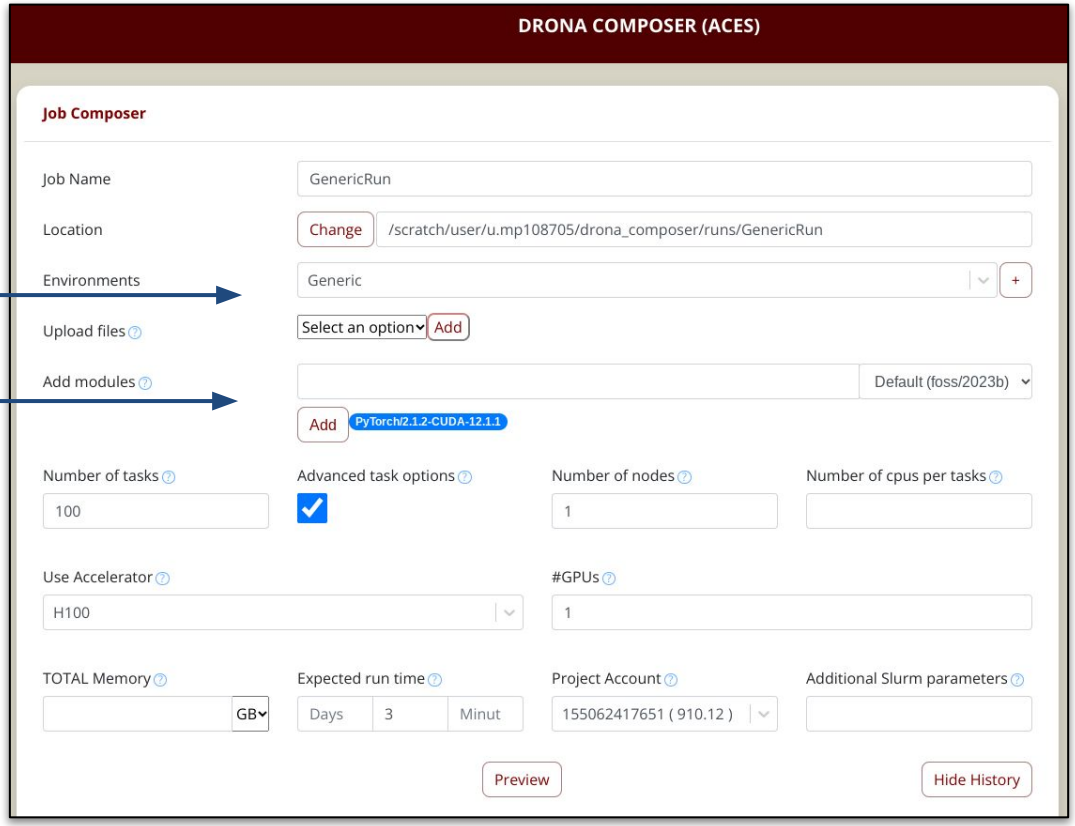
Portal: Drona Composer GUI



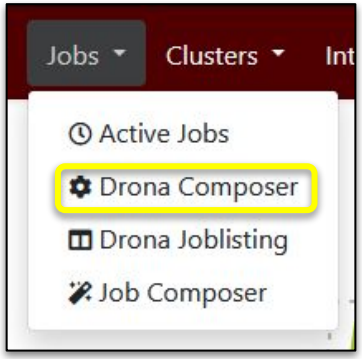
Generic environment to create custom Slurm batch job

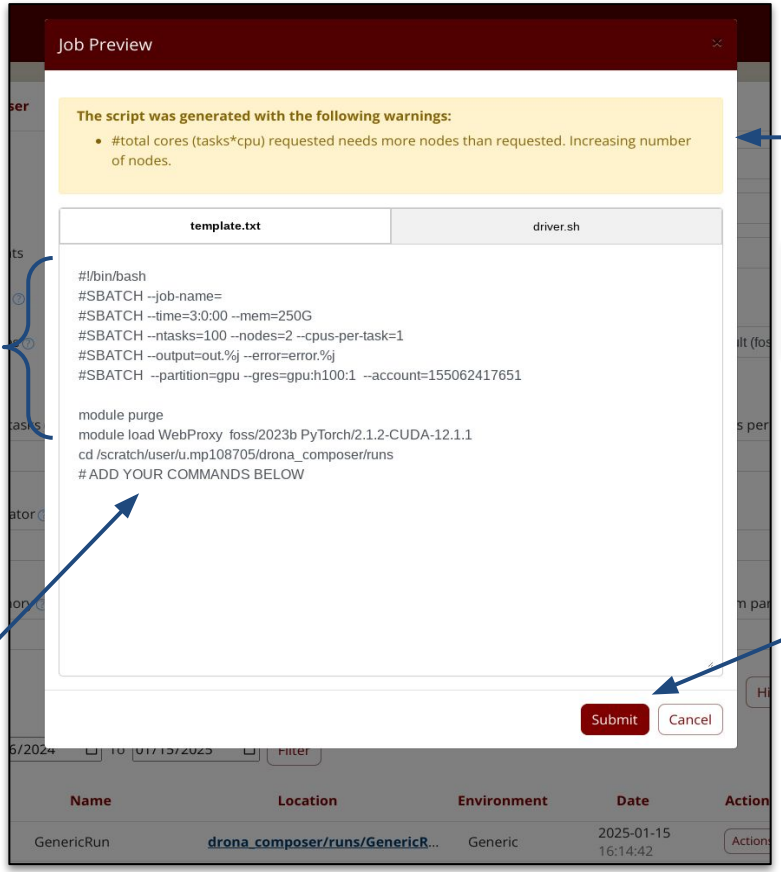
Add modules for job

Fill out info that Slurm would need



Portal: Drona Composer GUI





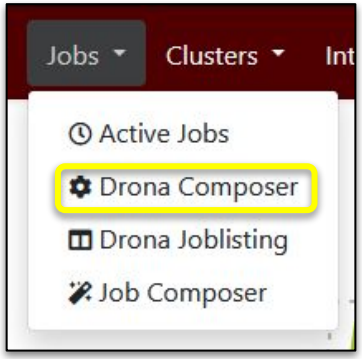
Pane with Input specific warnings

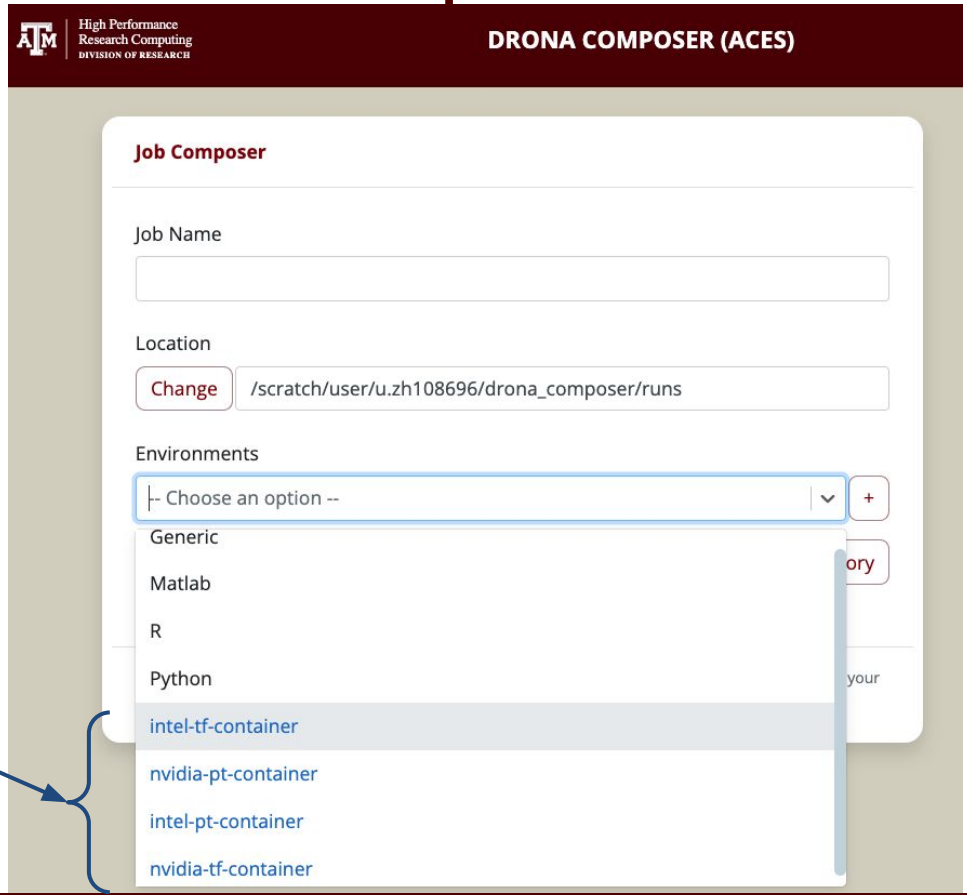
Generated Slurm directives and module loads

Add commands directly into editable window

Hit “Submit”

Portal: Drona Composer GUI





PyTorch and TensorFlower Containers

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

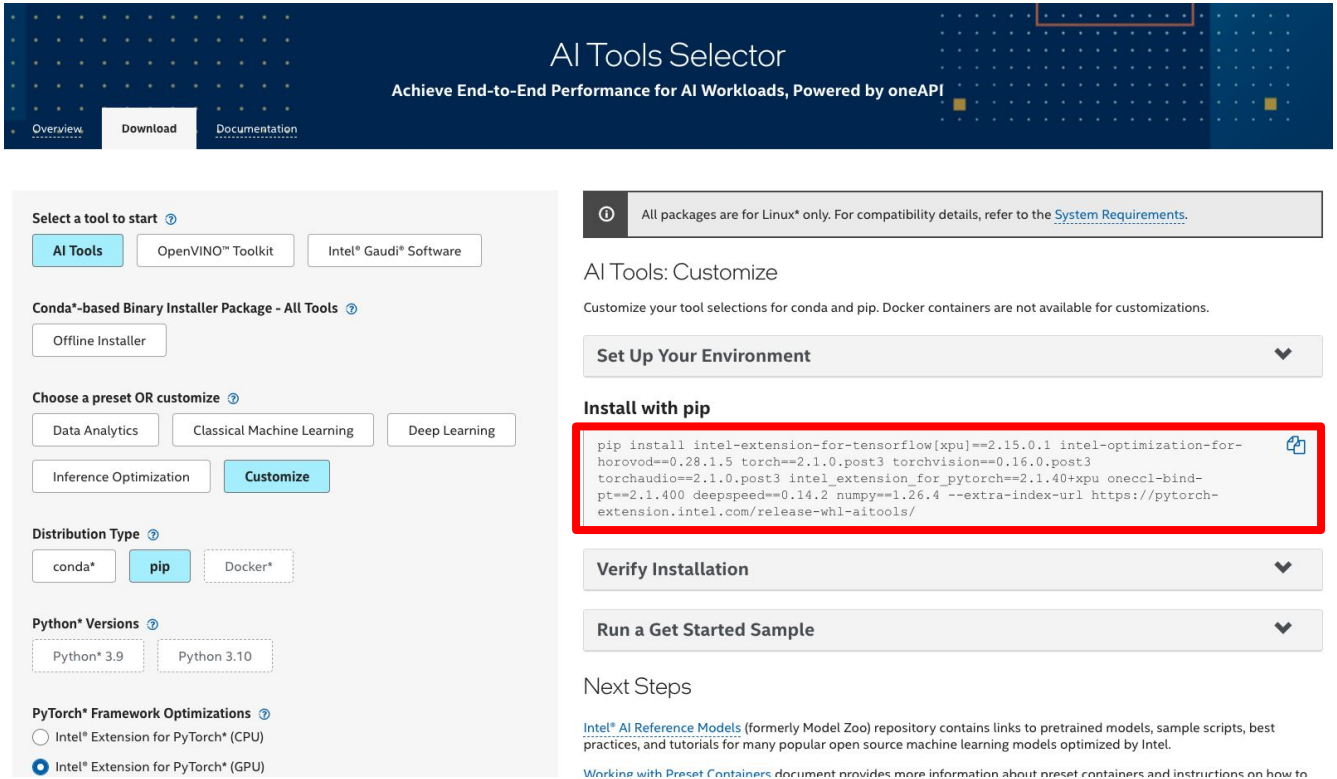
Intel® AI Analytics Toolkit
Accelerate machine learning & data science pipelines end-to-end with optimized DL frameworks & high-performing Python libraries

OpenVINO™

Intel® Distribution of OpenVINO™ Toolkit
Deploy high performance inference & applications from edge to cloud

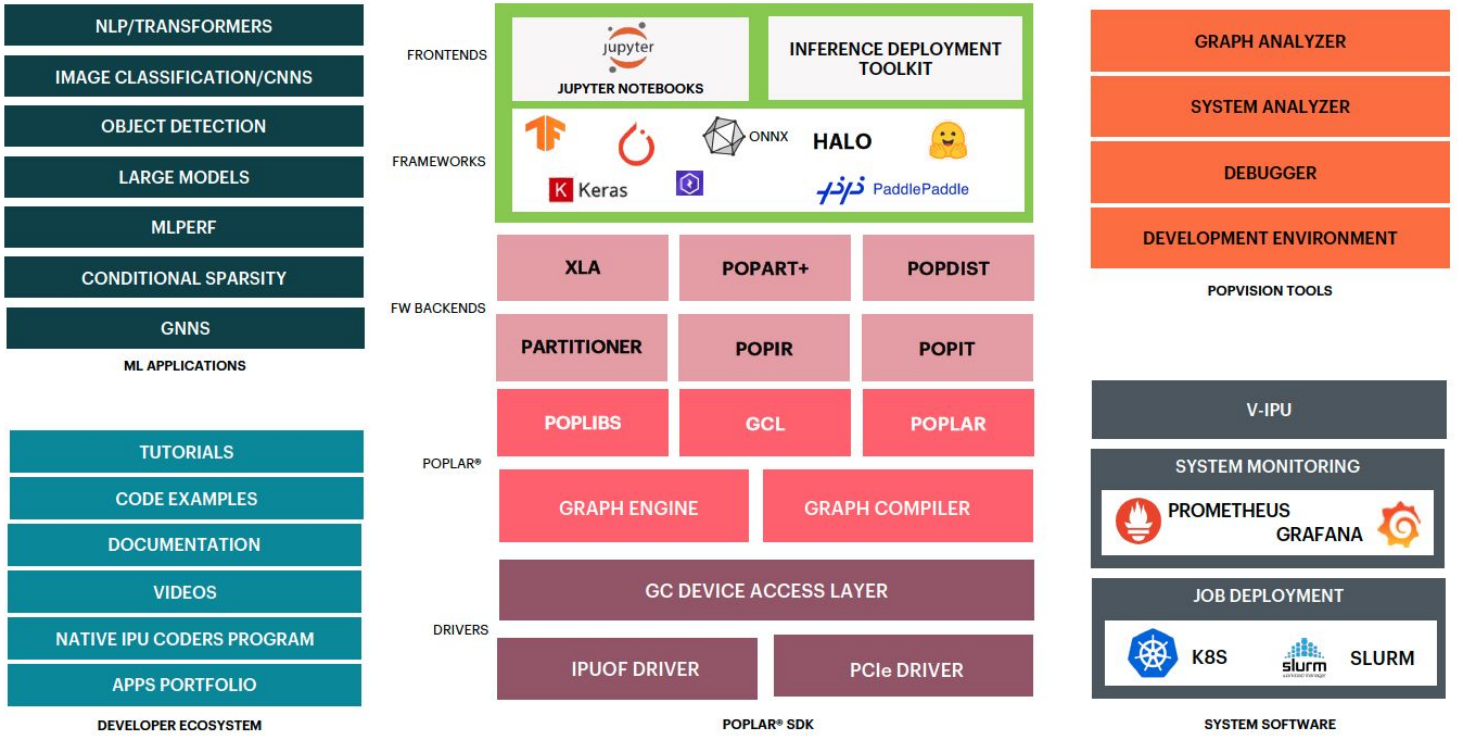
(Source: Intel)

Intel® AI Tools Selector



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

Graphcore Software Stack



Source: Graphcore

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/

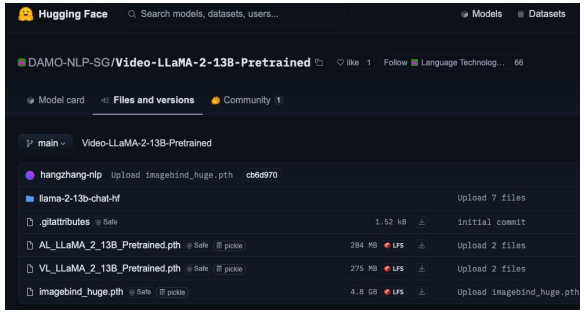
Hugging Face on ACES

ACCELERATING COMPUTING FOR EMERGING SCIENCES

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

    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
- …

```
[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
```

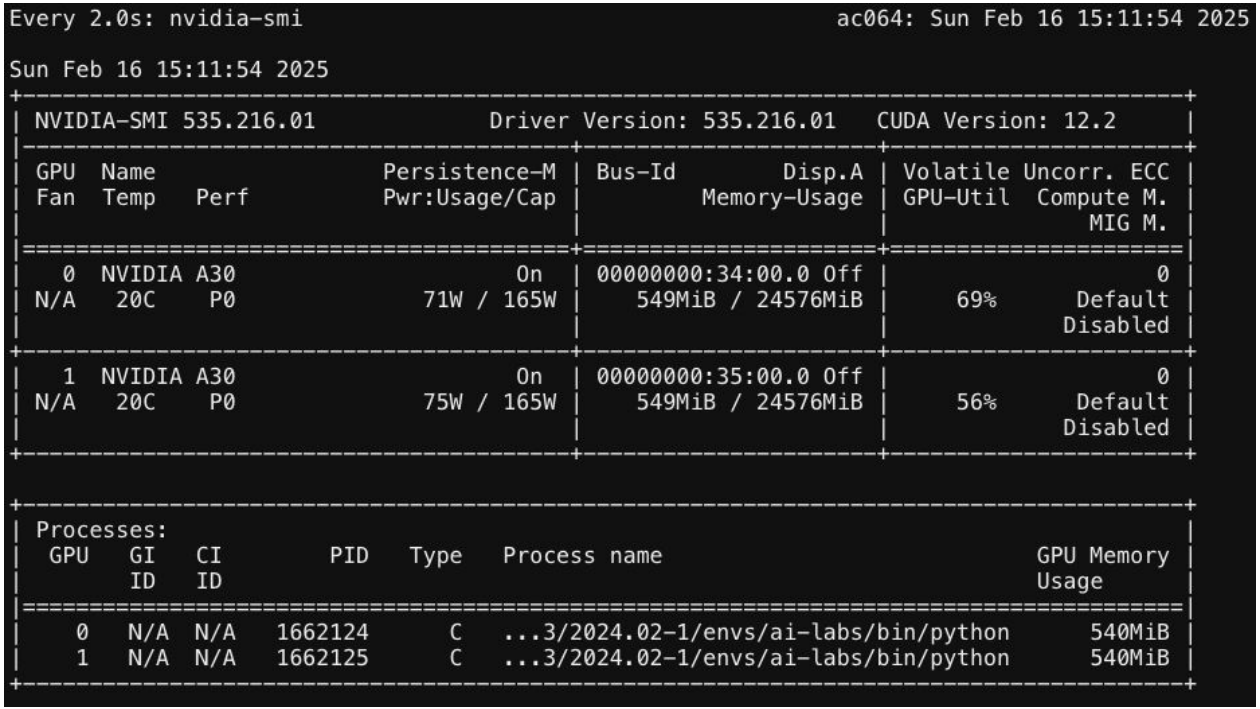
```
(hgf-env) [u.zh108696@aces-login3 huggingface]$ pip install transformers datasets tokenizers
Looking in indexes: https://pypi.org/simple, https://pypi.ngc.nvidia.com
Collecting transformers
  Downloading transformers-4.48.0-py3-none-any.whl.metadata (44 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━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

```python
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

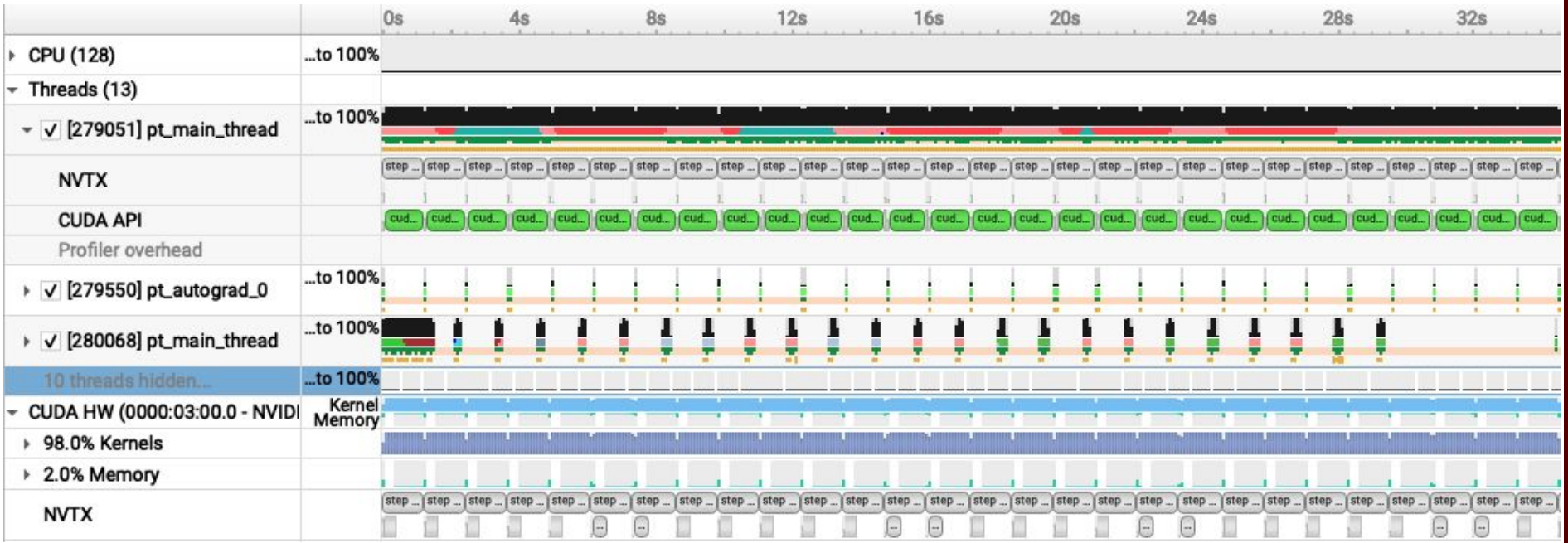
num_workers=0

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

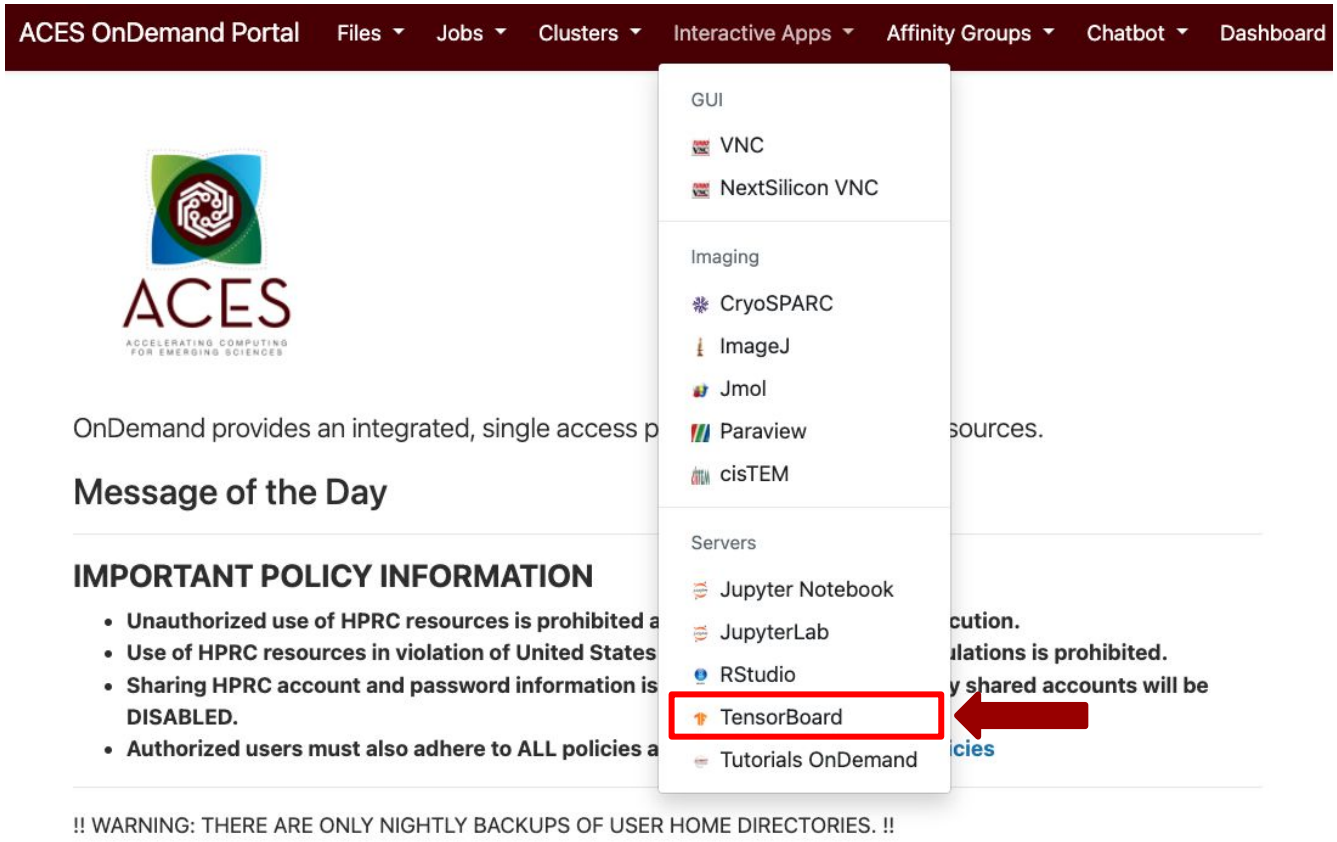
Originally developed for **TensorFlow**, it now supports **PyTorch, JAX, and standalone logging**.

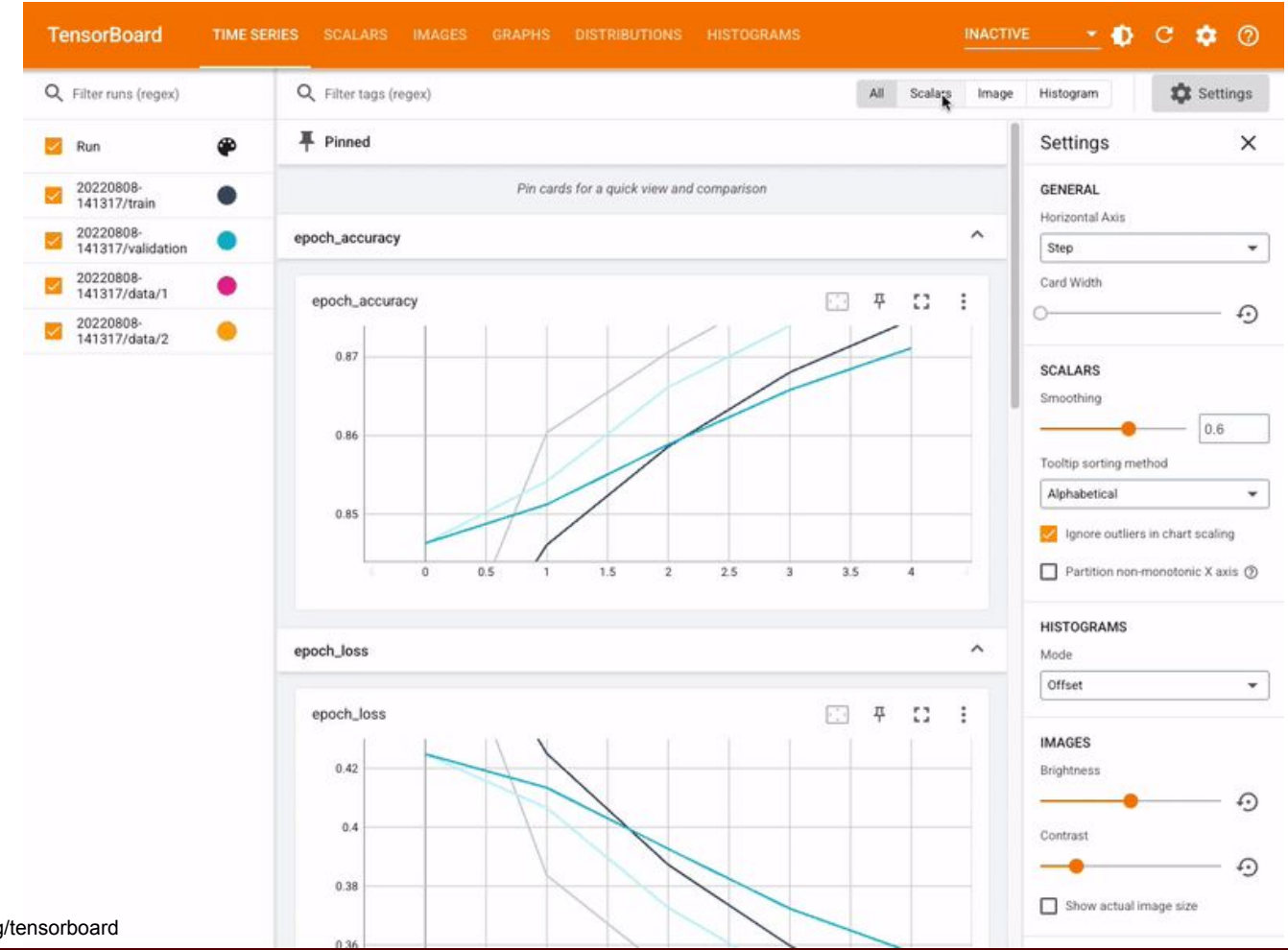
What Can TensorBoard Do?

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





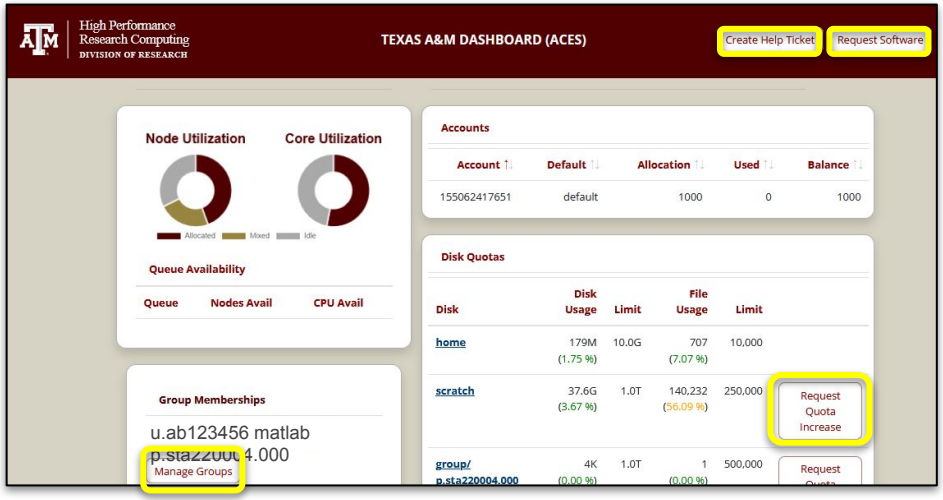
https://www.tensorflow.org/tensorboard

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!

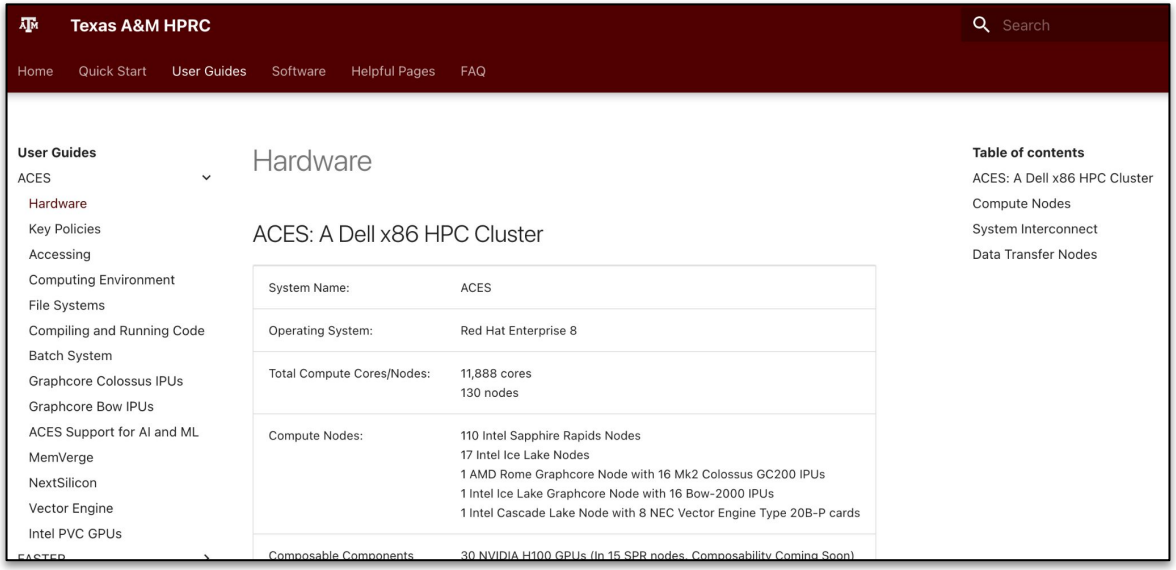
HPRC Survey

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



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

Texas A&M HPRC

@TexasAMHPRC · 1.35K subscribers · 107 videos

The Texas A&M HPRC YouTube channel assists HPR

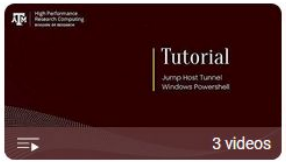
hprc.tamu.edu and 1 more link

Subscribe

https://www.youtube.com/texasamhprc

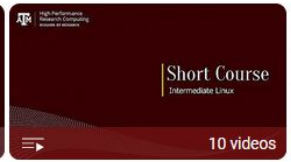
HOME VIDEOS PLAYLISTS COMMUNITY CHANNELS ABOUT

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

Accelerator Training

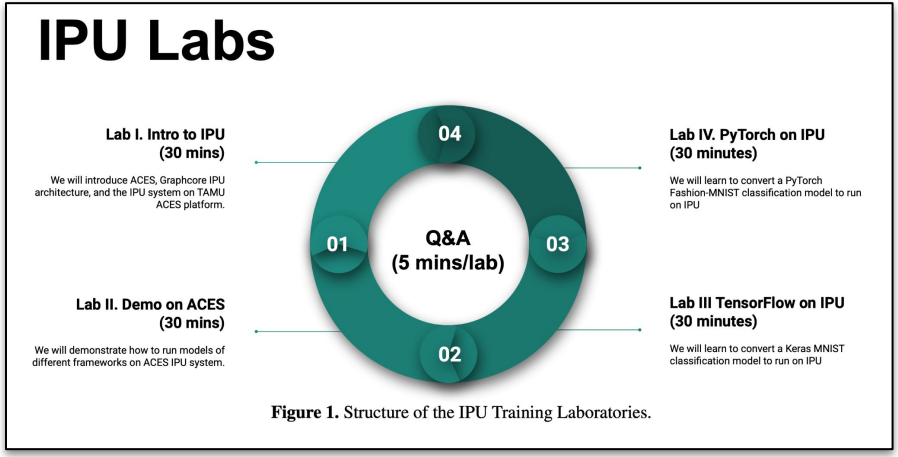
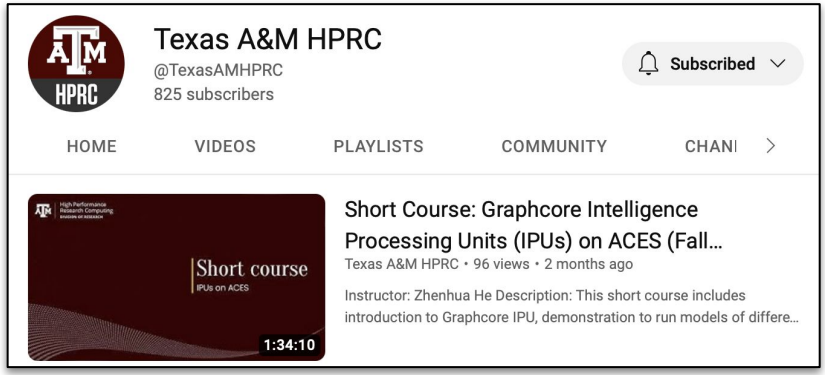


Figure 1. Structure of the IPU Training Laboratories.



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

- PVCs - Mar 4th - https://hprc.tamu.edu/training/aces_intel_pvc.html
- IPUs - 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

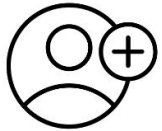
Allocations Management

ACCESS

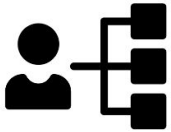
You can get allocations on both **ACES** and **FASTER** through *ACCESS*.

NAIRR Pilot

ACES can also be allocated through *NAIRR*.



CREATE ACCOUNT



SELECT OPPORTUNITY

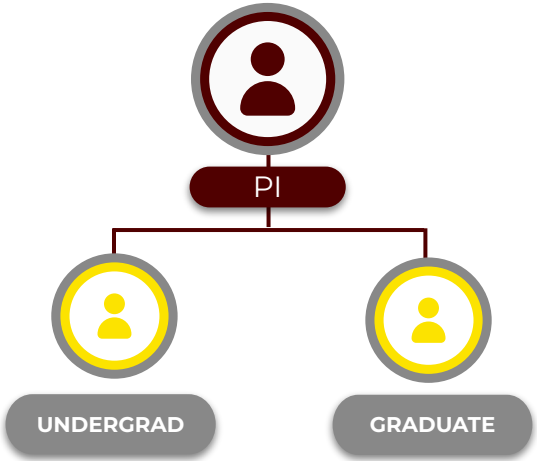
REQUEST ALLOCATION

RECEIVE CREDITS

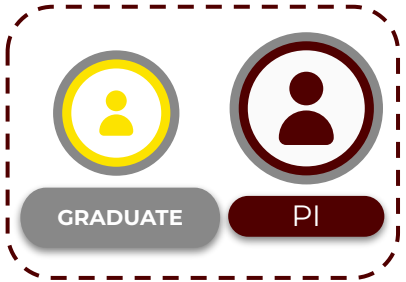
EXCHANGE CREDITS

Getting on ACES

- Using 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.



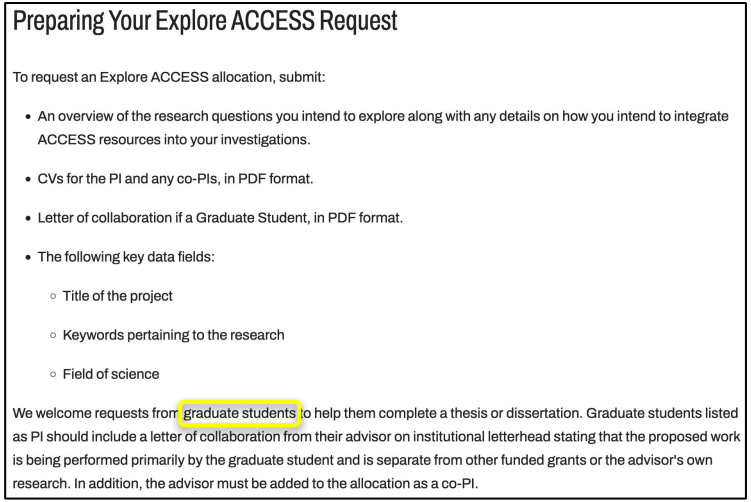
PIs can apply for an account and sponsor accounts for their researchers.



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

ACCESS Allocations Opportunities

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



Preparing Your Explore ACCESS Request

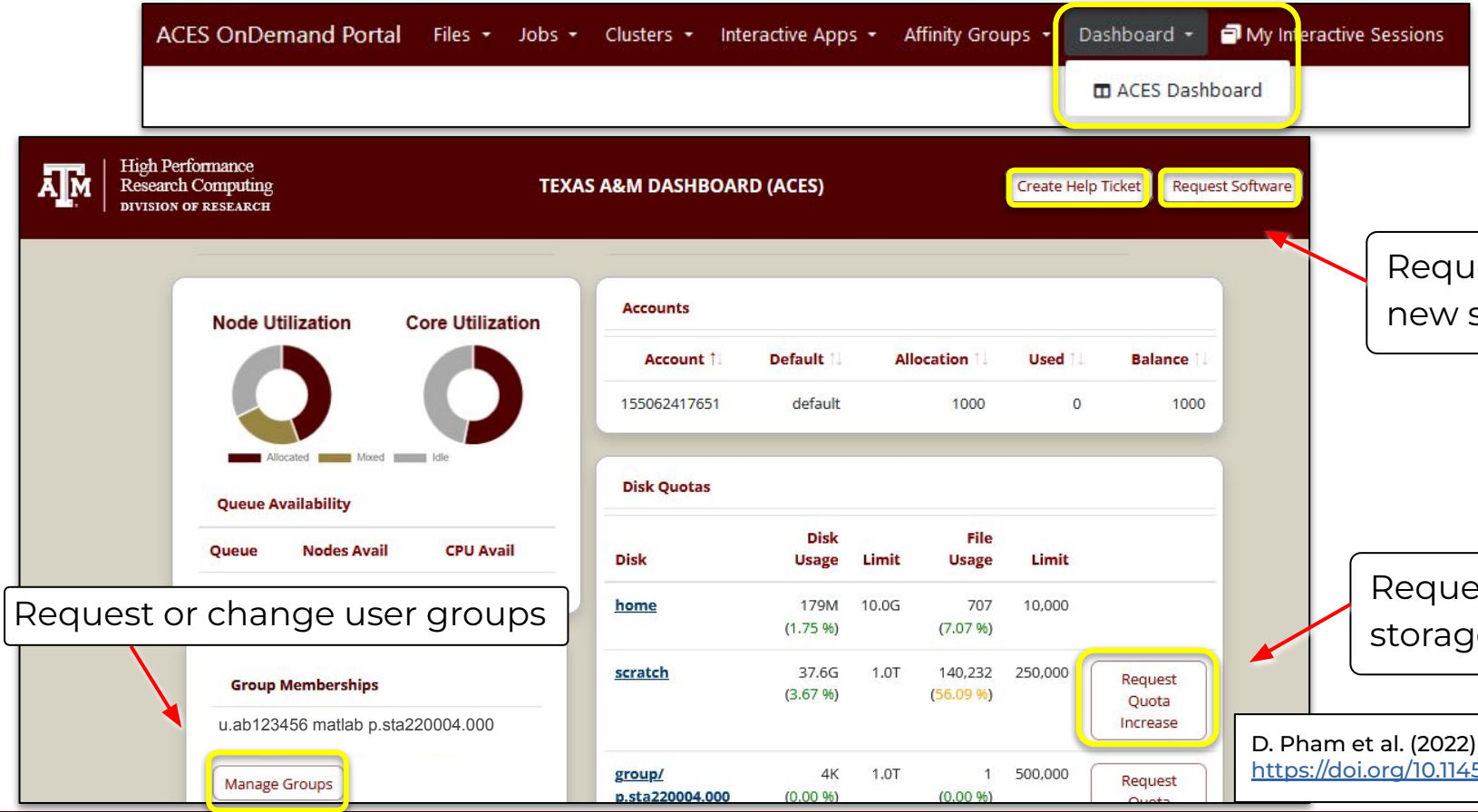
To request an Explore ACCESS allocation, submit:

- 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

ACES Dashboard



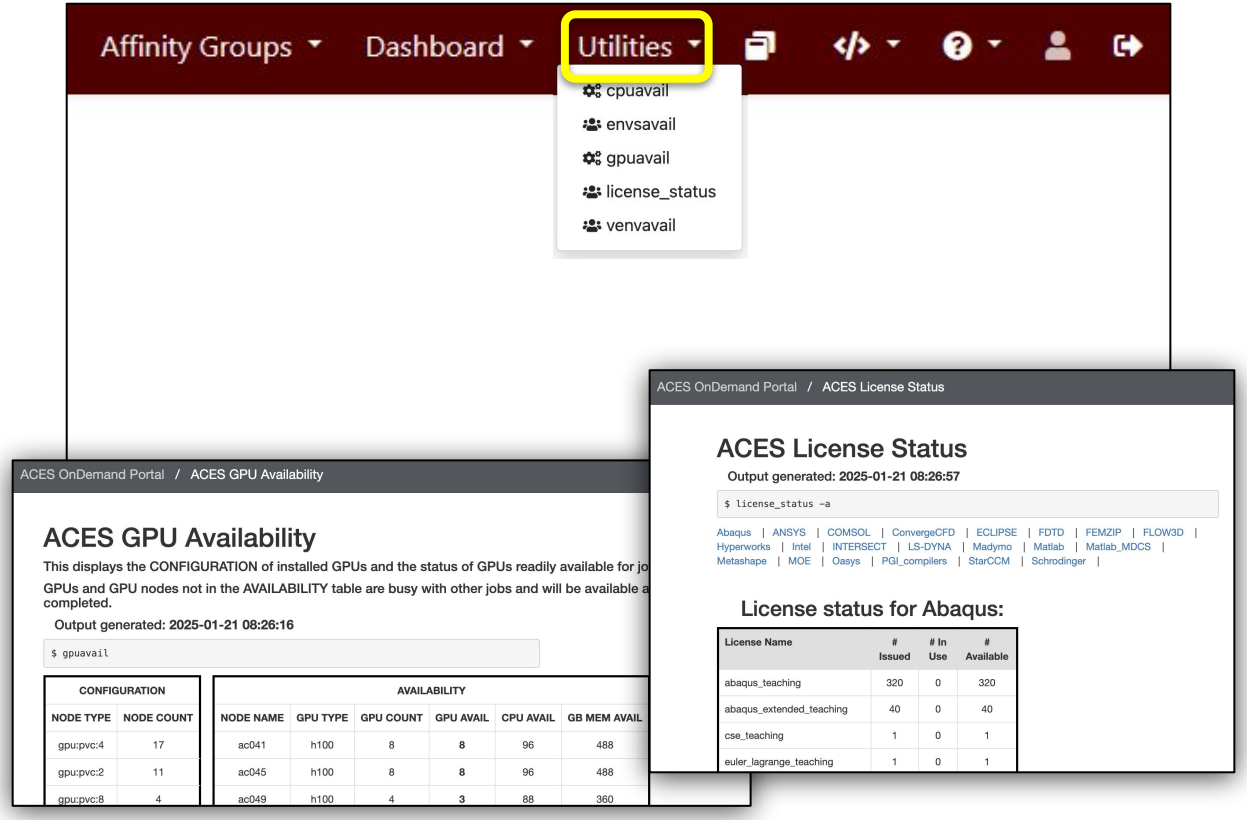
Request help or new software

Request or change user groups

Request higher storage limits

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

Utilities



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