

HIGH PERFORMANCE RESEARCH COMPUTING

ACES: AI/ML on Intel PVC GPUs

10/15/2024

Zhenhua He



High Performance
Research Computing
DIVISION OF RESEARCH



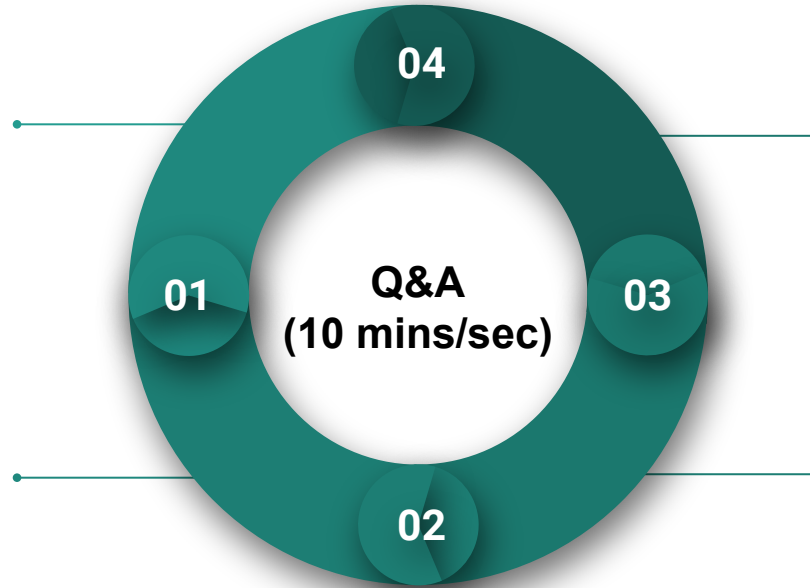
Outline

Section I. Intro to PVCs

We will introduce Intel's PVC, its architecture, and the PVC GPUs on the TAMU ACES platform.

Section II. Demo on ACES

We will demonstrate how to run models of different frameworks with PVC GPUs on the ACES system.



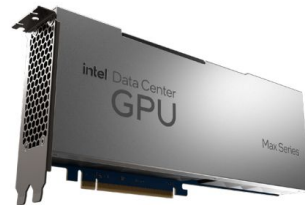
Section IV. TensorFlow on PVC

Students will learn how to convert a TensorFlow image classification model to run on a PVC GPU.

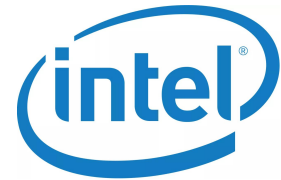
Section III PyTorch on PVC

Students will learn how to convert a PyTorch image classification model to run on a PVC GPU.

Lab I. Introducing Intel PVC GPUs on ACES



Intel Data Center GPU Max Series PCIe Card



NSF ACES

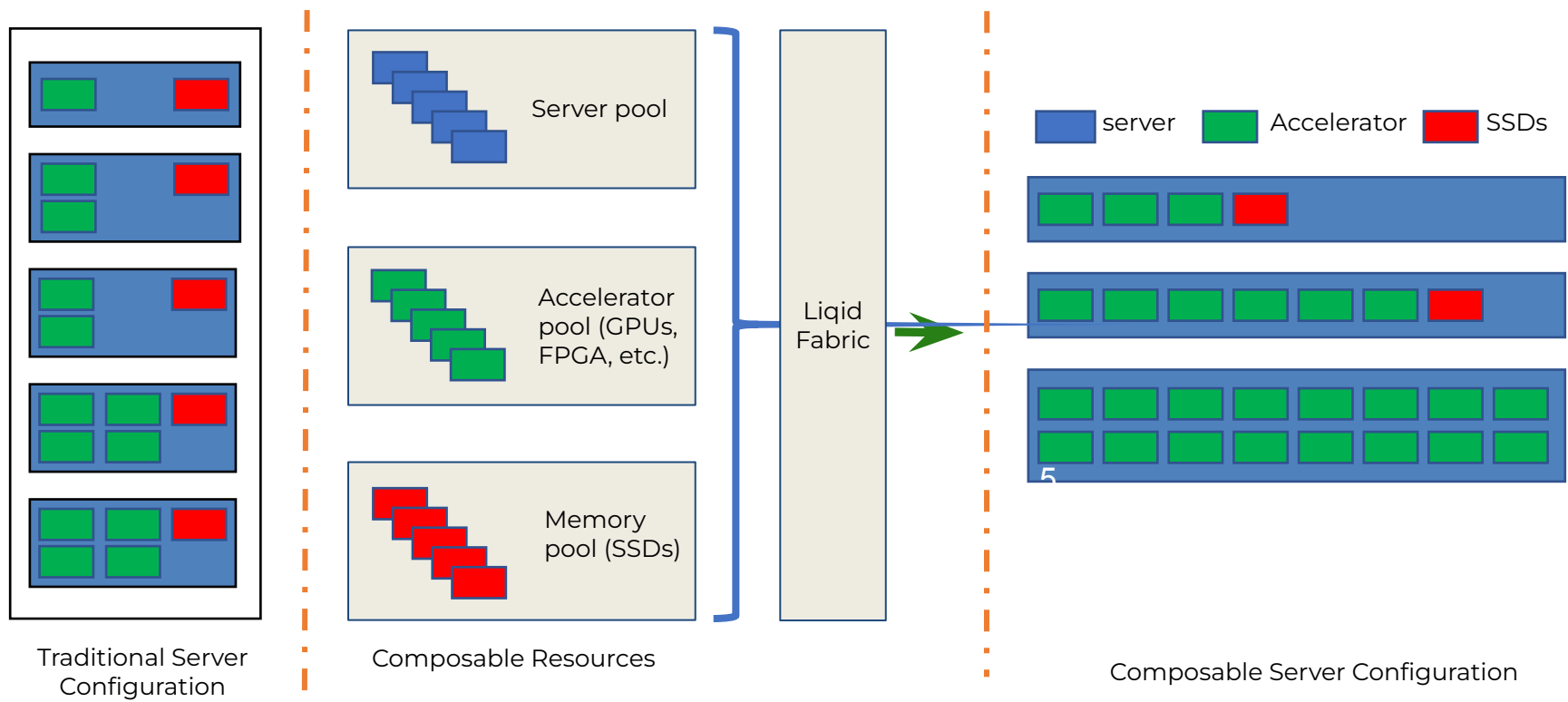
Accelerating Computing for Emerging Sciences

Our Mission:

- 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.



Design: Composability at the Hardware Level



ACES Accelerators

Component	Quantity	Description
Graphcore IPU	32	16 Colossus GC200 IPU, 16 Bow IPU. 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 Agilinx 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/>

Intel Max GPU 1100

- 1 tile/stack per card
- 56 X^e cores, 448 execution units (8 per core)
- 300W PCIe Gen5 x16 card
- 48GB HBM2e memory
- 1.2 TB/s memory bandwidth
- 22 TF FP64 peak performance



Intel Data Center GPU Max Series PCIe Card

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>)



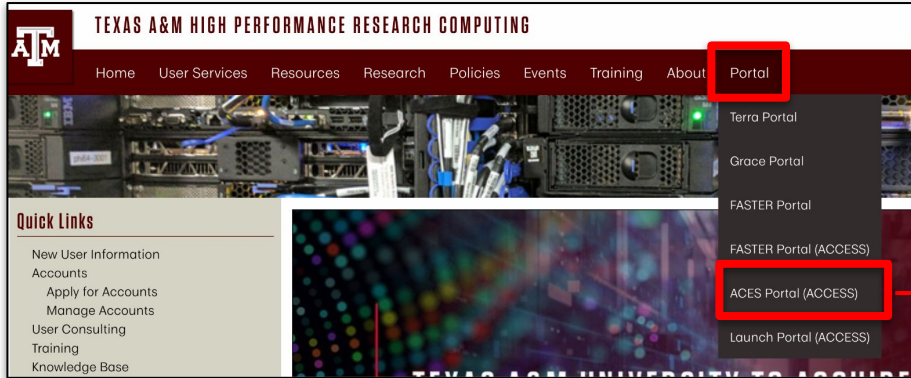
Shared Data Directories on ACES

- Datasets: ImageNet datasets for PyTorch and TensorFlow
`/scratch/data/pytorch-computer-vision-datasets`
`/scratch/data/tensorflow-computer-vision-datasets`
- Models: Intel AI models
`/scratch/data/intel-ai-models`
- Containers
`/scratch/data/containers/intel-deep-learning-2023.2-py3.10-perms.sif`
- Shared Python Virtual environment
`/sw/hprc/sw/Python/virtualenvs/intelpython/2024.1.0_814/intel-ai-python-env`
- The `setvars.sh` script sets environment variables for use with the oneAPI toolkits
`/sw/hprc/sw/oneAPI/<version>/setvars.sh`

Resources

- [Texas A&M High Performance Research Computing \(HPRC\)](#)
- [ACES Quick Start Guide](#)
- [ACES Portal \(ACCESS\)](#)
- [ACCESS Documentation](#)
- [HPRC YouTube Channel](#)
- help@hprc.tamu.edu

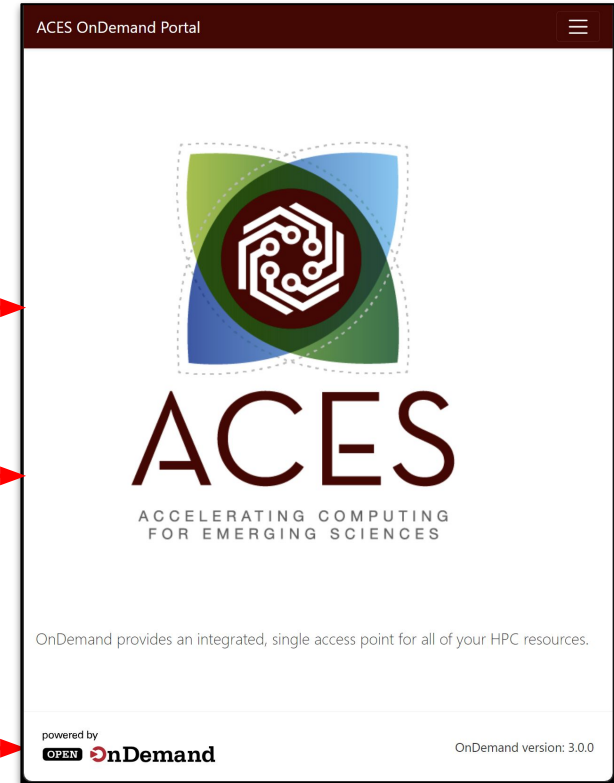
ACES Portal



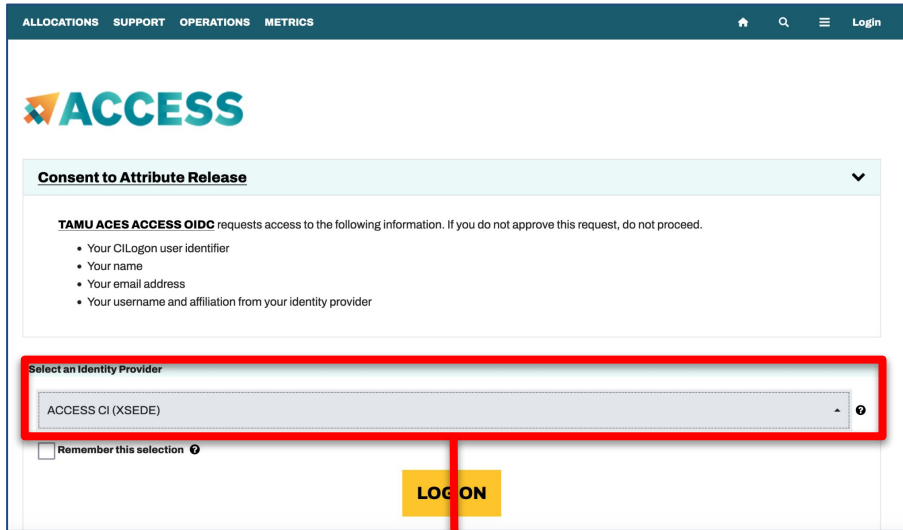
ACES Portal portal-aces.hprc.tamu.edu
is the web-based user interface for the ACES cluster

[HPRC Portal YouTube tutorials](#)

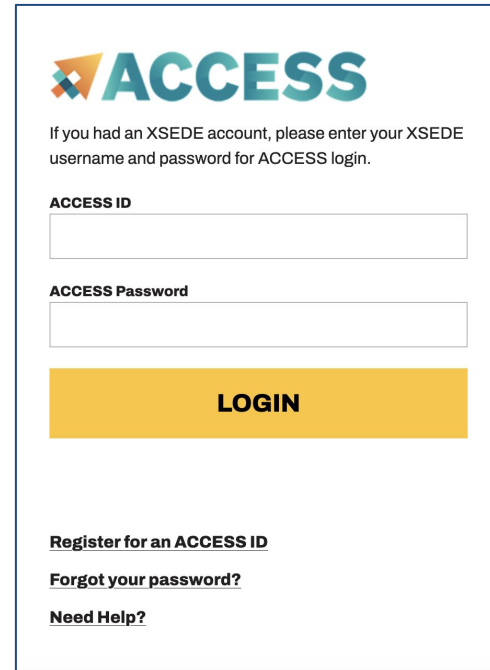
Open OnDemand (OOD) is an advanced web-based graphical interface framework for HPC users



Accessing ACES via the Portal (ACCESS)



Select the Identity Provider appropriate for your account.



Log-in using your ACCESS or institutional credentials.

Shell Access via the Portal

ACES OnDemand Portal Files Jobs Clusters Interactive Apps Affinity Groups Dashboard

>_aces Shell Access

Get a shell terminal right in your browser

ACES
ACCELERATING COMPUTING
FOR EMERGING SCIENCES

```
Host: login.aces Theme: Default
*****
This computer system and the data herein are available only for authorized
purposes by authorized users. Use for any other purpose is prohibited and may
result in disciplinary actions or criminal prosecution against the user. Usage
may be subject to security testing and monitoring. There is no expectation of
privacy on this system except as otherwise provided by applicable privacy laws.
Refer to University SAP 29.01.03.M0.02 Acceptable Use for more information.
*****

Last login: Wed Mar 13 09:55:42 2024 from 10.71.1.6

=====
Texas A&M University High Performance Research Computing

Website:          https://hprc.tamu.edu
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/
*****

**** ACES Update, March 7 ****

The pvc queue has been updated with a new set of nodes with 2x, 4x, and 8x PVCs.

!! 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 moio command.
Your current disk quotas are:
Disk          Disk Usage   Limit   File Usage   Limit
/home/u.zh108696 5.4G       10.0G   3148         10000
/scratch/user/u.zh108696 439.2G    1.0T    1169787      2000000
Type 'showquota' to view these quotas again.
[u.zh108696@aces-login2 ~]$
```

PVC Slurm Nodes Status Check

- View the pvc nodes and number of GPUs

```
$ pestat -p pvc -G
```

- View more details of the pvc node features

```
$ show_pvc_features
```

HOSTNAME	AVAIL_FEATURES	GRES	STATE
ac010	gen4_fabric	gpu:pvc:4	mixed
ac011	gen4_fabric	gpu:pvc:4	mixed
ac012	gen4_fabric	gpu:pvc:4	mixed
ac013	gen4_fabric	gpu:pvc:4	mixed
ac023	gen4_fabric	gpu:pvc:4	idle
ac024	gen4_fabric	gpu:pvc:8	mixed
ac025	gen4_fabric	gpu:pvc:4	mixed
ac026	gen5_fabric	gpu:pvc:4	drained*
ac028	gen5_fabric	gpu:pvc:4	drained*
ac030	gen5_fabric	gpu:pvc:4	drained*
ac032	gen5_fabric	gpu:pvc:4	drained*
ac034	gen5_fabric	gpu:pvc:4	drained*
ac039	gen5_fabric	gpu:pvc:4	drained*
ac050	gen5_nonfabric	gpu:pvc:2	drained*
ac051	gen5_nonfabric	gpu:pvc:2	drained*
ac062	gen4_fabric	gpu:pvc:4	drained
ac068	gen4_fabric	gpu:pvc:8	mixed
ac078	gen4_fabric	gpu:pvc:4	mixed
ac079	gen4_fabric	gpu:pvc:4	mixed
ac081	gen5_fabric,xelink4	gpu:pvc:4	drained*
ac082	gen5_fabric,xelink2	gpu:pvc:2	drained*
ac083	gen5_fabric	gpu:pvc:2	drained*
ac085	gen5_fabric,xelink4	gpu:pvc:4	drained*
ac086	gen5_fabric,xelink2	gpu:pvc:2	drained*
ac087	gen5_fabric,xelink2	gpu:pvc:2	drained*
ac089	gen5_fabric,xelink4	gpu:pvc:4	drained*
ac094	gen5_fabric,xelink2	gpu:pvc:2	drained*
ac095	gen5_fabric,xelink2	gpu:pvc:2	drained*
ac097	gen5_fabric,xelink2	gpu:pvc:2	reserved
ac099	gen5_fabric,xelink4	gpu:pvc:4	mixed
ac100	gen5_fabric,xelink2	gpu:pvc:2	mixed
ac101	gen5_fabric	gpu:pvc:4	mixed
ac102	gen5_fabric	gpu:pvc:4	mixed
ac103	gen5_fabric,xelink2	gpu:pvc:2	mixed

Copy the Materials to Personal Directory

- Navigate to your personal scratch directory

```
$ cd $SCRATCH
```

- Files for this course are located at

```
/scratch/training/aces_pvc_tutorial
```

Make a copy in your personal scratch directory

```
$ cp -r /scratch/training/aces_pvc_tutorial $SCRATCH
```

- Enter this directory (your local copy)

```
$ cd $SCRATCH/aces_pvc_tutorial
```


Lab II. Using PVCs on ACES



Environment Setup

Option 1. Use a Shared Python Virtual Environment

```
# load all the necessary modules
module purge
module load WebProxy
module load intelpython/2024.1.0_814
module load intel/2023.07

# activate the shared virtual environment
source /sw/hprc/sw/Python/virtualenvs/intelpython/2024.1.0_814/
intel-ai-python-env/bin/activate

# sets environment variables for use with the oneAPI toolkits
source /sw/hprc/sw/oneAPI/2024.2/setvars.sh
```

`$SCRATCH/aces_pvc_tutorial/pytorch/pt_demo.slur`

Environment Setup

AI Tools Selector



Option 2. Create a Python Virtual Environment

```
# clean up and load Anaconda
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
```

Run PyTorch ResNet50 model

- We have prepared a Slurm job file (*pt_demo.slurm*) to run the PyTorch ResNet50 model. Submit the job using the command

```
$ cd pytorch/
```

```
$ sbatch pt_demo.slurm
```

Run Tensorflow ResNet50 Model

- We have prepared a Slurm job file (*tf_demo.slurm*) to run the Tensorflow ResNet50 model. Submit the job using the command

```
$ cd ..
```

```
$ cd tensorflow/
```

```
$ sbatch tf_demo.slurm
```

Lab III. PyTorch on PVC



1. Import Intel Extension for PyTorch

Intel Extension for PyTorch is a Python package for extending PyTorch models to run on an Intel platform.

Add the following import statement to the beginning of your script:

```
import intel_extension_for_pytorch as ipex
```

2. Move the Model and Criterion to “xpu”

```
model = model.to("xpu")
```

```
criterion = criterion.to("xpu")
```


3. Apply the “ipex optimize” Function

Apply the ipex optimize function against the model and optimizer objects.

```
model, optimizer = ipex.optimize(model, optimizer=optimizer,  
dtype=torch.bfloat16)
```

4. Move the Data and Target to “xpu”

In the training loop,

```
data = data.to("xpu")
```

```
target = target.to("xpu")
```

5. Use Auto Mixed Precision (AMP)

Use automatic mixed-precision (AMP) with BFloat16 data type with the *torch.xpu.amp.autocast* context manager

```
with torch.xpu.amp.autocast(enabled=True, dtype=torch.bfloat16):
```

Hands-on Session

- Navigate to the PyTorch exercises directory

```
$ cd $SCRATCH/aces_pvc_tutorial/pytorch/exercises
```

- Open the exercise file (*cifar10_pvc_todo.py*) with your preferred editor (e.g. vim) or the file editor of the OnDemand portal.
- Complete the **#Todos** in the *cifar10_pvc_todo.py* file.
- Modify the Slurm job file (*pt_cifar10_pvc.slurm*) and submit your job.

```
$ sbatch pt_cifar10_pvc.slurm
```

Lab IV. TensorFlow on PVC



Install Intel Extension for Tensorflow

The Intel Extension for Tensorflow is based the on Tensorflow PluggableDevice interface to bring Intel XPU (GPU, CPU, etc) devices into Tensorflow.

To check the version, add import statement to the beginning of your script:

```
import intel_extension_for_tensorflow as itex
print(itex.__version__)
```

The default device will be Intel GPU after installing `intel-extension-for-tensorflow`

Source: Intel presentation at ACES Workshop

No Code Changes are Needed!



Credit: Bing Chat Enterprise

Hands-on Session

- Navigate to the TensorFlow exercises directory

```
$ cd $SCRATCH/aces_pvc_tutorial/tensorflow/exercises
```

- Open the exercise file (*cifar10_pvc.py*) with your preferred editor (e.g. vim) or the file editor of the OnDemand portal.
- Read through the code to verify that there are no code changes
- Modify the Slurm job file (*tf_cifar10_pvc.slurm*) and submit your job.

```
$ sbatch tf_cifar10_pvc.slurm
```

```
##### TODO #####  
#uncomment below lines to run the cifar-10 exercise  
cd $SCRATCH/aces_pvc_tutorial/tensorflow/exercises  
python cifar10_pvc.py  
##### END TODO #####
```


PVC Monitoring Tools

- View the pvc nodes and number of GPUs

```
$ pestat -p pvc -G
```

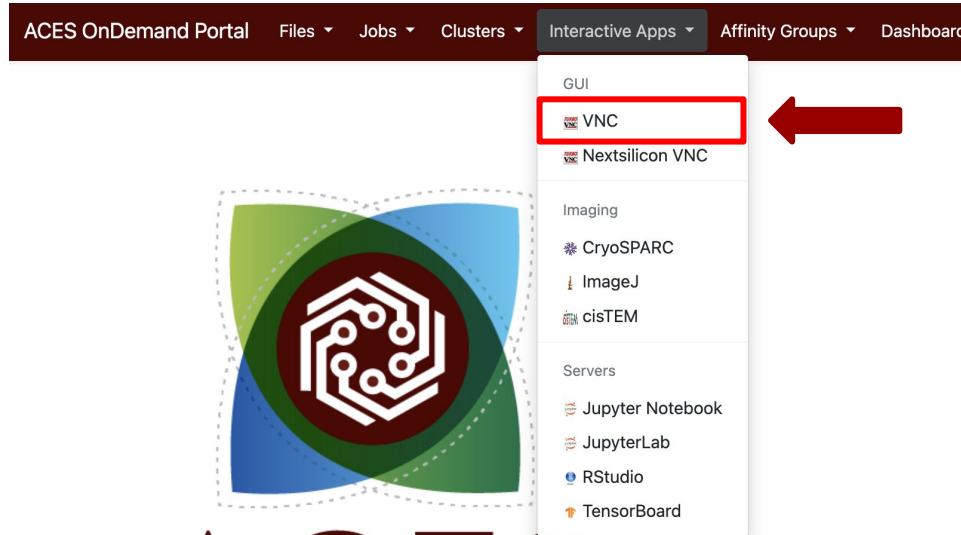
- Monitor the system activity

```
$ watch -n 5 sysmon
```

- Intel XPU manager

```
$ watch -n 5 xpumcli stats -d <device index>
```

Start a VNC job



The screenshot shows the ACES OnDemand Portal navigation bar with the following items: ACES OnDemand Portal, Files, Jobs, Clusters, Interactive Apps, Affinity Groups, and Dashboard. The 'Interactive Apps' dropdown menu is open, listing the following categories and items:

- GUI
 - VNC (highlighted with a red box and a red arrow pointing to it)
 - Nextsilicon VNC
- Imaging
 - CryoSPARC
 - ImageJ
 - cisTEM
- Servers
 - Jupyter Notebook
 - JupyterLab
 - RStudio
 - TensorBoard

ACES

ACCELERATING COMPUTING
FOR EMERGING SCIENCES

VNC Form

[Home](#) / [My Interactive Sessions](#) / VNC

Interactive Apps

- GUI
- VNC**
- NextSilicon VNC
- Imaging
 - CryoSPARC
 - ImageJ
 - Jmol
 - Paraview
 - cisTEM
- Servers
 - Jupyter Notebook
 - JupyterLab

VNC

This app will launch a **VNC** job on **ACES** for remote visualization.

Node type

Intel GPU Max (PVC)

- select a non-CPU node type only if your software supports the Accelerator

Number of GPUs

1

- Run the **gpuavail** command on the terminal (Clusters -> _aces Shell Access) to see the current GPU configuration and availability

Number of hours (max 48)

1

Number of cores (max 96)

3

Total GB Memory (max 488)

5

Fields:

Node Type: Intel GPU Max (PVC)

Number of GPUs: 1

Number of hours: 1

Number of cores: 3

Total memory (GB): 5

Launch VNC

Session was successfully created. ✕

[Home](#) / My Interactive Sessions

Interactive Apps

- GUI
- VNC
- NextSilicon VNC
- Imaging
- CryoSPARC
- ImageJ
- Jmol
- Paraview

VNC (272475) 1 node | 1 core | Running

Host: >_ac011 ✕ Delete

Created at: 2024-10-10 22:11:52 CDT

Time Remaining: 59 minutes

Session ID: [afb00e2-65db-4c30-81ae-ae313bcf1b32](#)

Compression 0 (low) to 9 (high) | Image Quality 0 (low) to 9 (high)

Launch VNC View Only (Share-able Link)

```
[u.zh108696@ac011 u.zh108696]$ cd aces_pvc_tutorial/pytorch/exercises/
[u.zh108696@ac011 exercises]$ ml purge
[u.zh108696@ac011 exercises]$ ml WebProxy
[u.zh108696@ac011 exercises]$ ml intelpython/2024.1.0_814
[u.zh108696@ac011 exercises]$ source /sw/hprc/sw/Python/virtualenvs/intelpython/2024.1.0_814/intel-ai-python-env/bin/activate
(intel-ai-python-env) [u.zh108696@ac011 exercises]$ ml intel/2023.07
(intel-ai-python-env) [u.zh108696@ac011 exercises]$ source /sw/hprc/sw/oneAPI/2024.2/setvars.sh
```

```
:: initializing oneAPI environment ...
bash: BASH_VERSION = 4.4.20(1)-release
args: Using "$@" for setvars.sh arguments:
```

```
:: advisor -- latest
:: ccl -- latest
:: compiler -- latest
:: dal -- latest
:: debugger -- latest
:: dev-utilities -- latest
:: dnnl -- latest
:: dpcpp-ct -- latest
:: dpl -- latest
:: ipp -- latest
:: ippcp -- latest
:: mkl -- latest
:: mpi -- latest
:: tbb -- latest
:: vtune -- latest
:: oneAPI environment initialized ::
```

```
(intel-ai-python-env) [u.zh108696@ac011 exercises]$
Display all 2271 possibilities? (y or n)
(intel-ai-python-env) [u.zh108696@ac011 exercises]$ python cifar10_pvc_solution.py > out.txt 2>&1 &
[1] 3444767
(intel-ai-python-env) [u.zh108696@ac011 exercises]$ watch sysmon
```

```
Every 2.0s: sysmon
```

```
=====
GPU 0: Intel(R) Data Center GPU Max 1100   PCI Bus: 0000:38:00.0
Vendor: Intel(R) Corporation   Driver Version: 1.3.27642   Subdevices: 0
EU Count: 448   Threads Per EU: 8   EU SIMD Width: 16   Total Memory(MB): 46679.2
Core Frequency(MHz): 1550.0 of 1550.0   Core Temperature(C): unknown
=====
```

```
Running Processes: 3
```

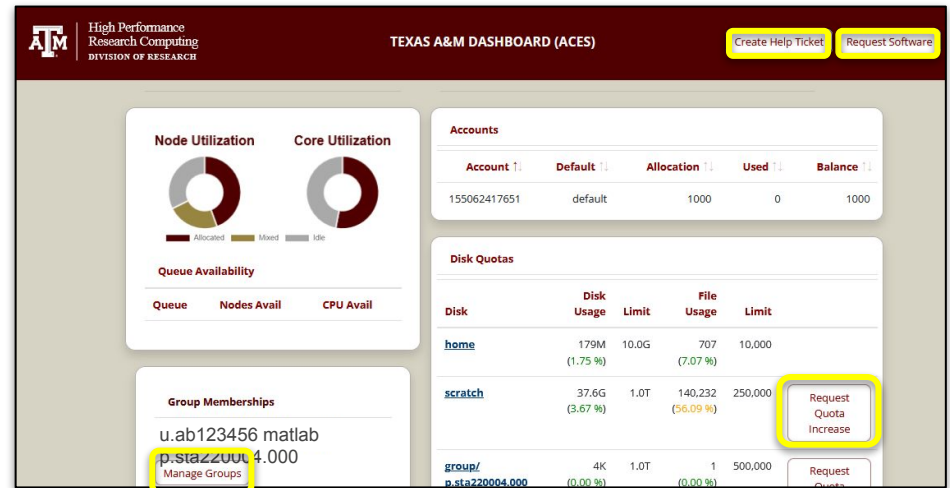
PID,	Device Memory Used(MB),	Shared Memory Used(MB),	GPU Engines,	Executable
4622,	1.8,	0.0,	COMPUTE,	/usr/bin/xpumd
3444767,	6556.4,	0.0,	COMPUTE;DMA,	python
3447277,	1.8,	0.0,	UNKNOWN,	sysmon

Need Help?

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

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

Remember the
Dashboard!



Need Help?

Help us help you -- tell us:

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



High Performance
Research Computing
DIVISION OF RESEARCH

Thank you
Questions?

Give us feedback on the class with this survey:
https://u.tamu.edu/hprc_shortcourse_survey



HPRC Survey