AI/ML Workflows on ACES Accelerators

Zhenhua He and Wesley Brashear

07/19/2024

ACES Workshop, Providence RI

ACES

ACCELERATING COMPUTING FOR EMERGING SCIENCES

TACC

TEXAS ADVANCED COMPUTING CENTER

High Performance Research Computing

DIVISION OF RESEARCH

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

Outline

- Introduction

  Introduce HPRC resources, ACES, and how to access ACES and AI/ML

- AI/ML on Nvidia GPUs

  Use Nvidia H100 to run a DL image classification model

- AI/ML on Graphcore IPUs

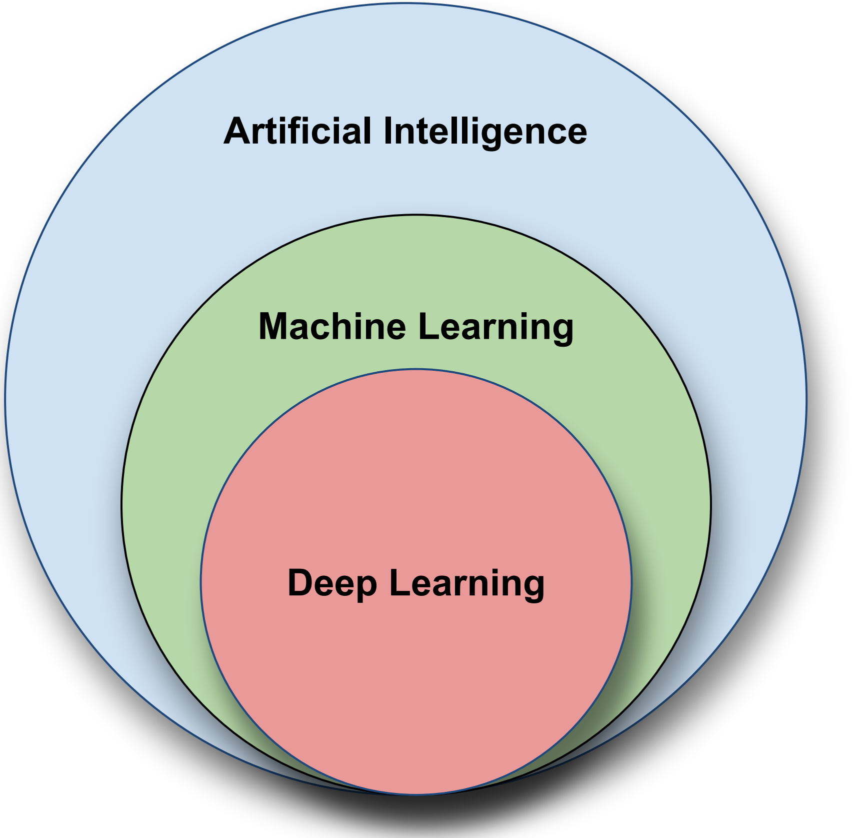
  Introduce IPU architecture and DL model porting

- AI/ML on Intel PVC GPUs

  Introduce PVC specs, run DL models on PVC GPUs

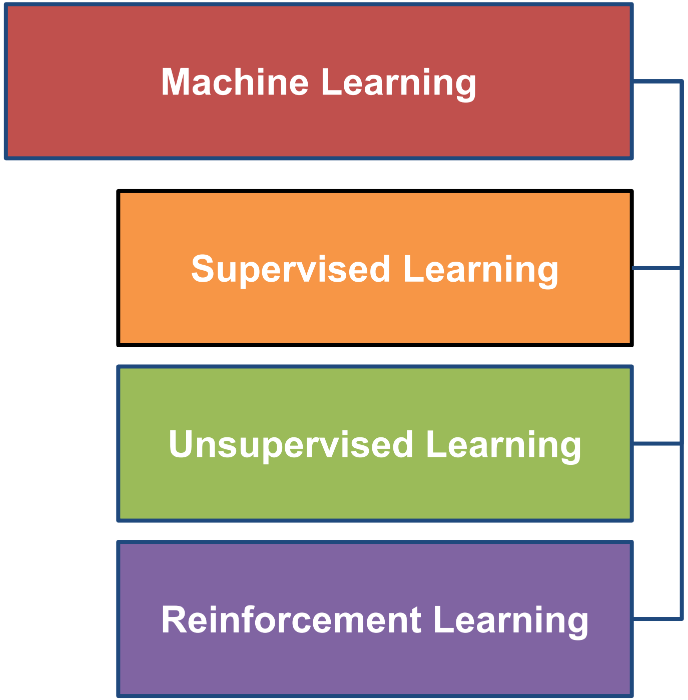
Relationship of AI, ML, and DL

- **Artificial Intelligence (AI)** is anything about man-made intelligence exhibited by machines.
- **Machine Learning (ML)** is an approach to achieve **AI**.
- **Deep Learning (DL)** is one technique to implement **ML**.

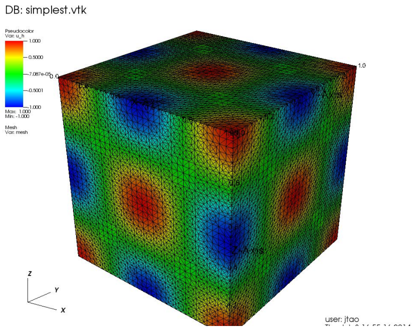


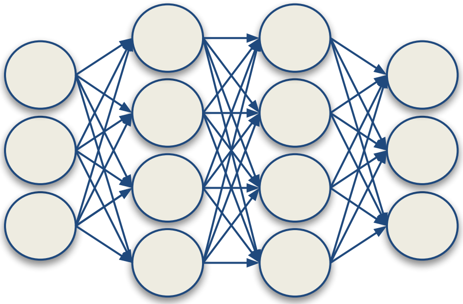
Types of ML Algorithms

- **Supervised Learning**
  - trained with labeled data; including regression and classification problems
- **Unsupervised Learning**
  - trained with unlabeled data; clustering and association rule learning problems.
- **Reinforcement Learning**
  - no training data; stochastic Markov decision process; robotics and business strategy planning.

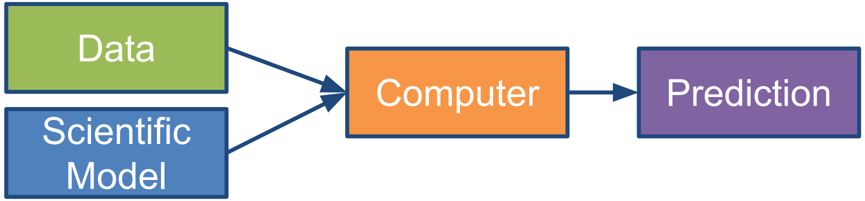


Machine Learning

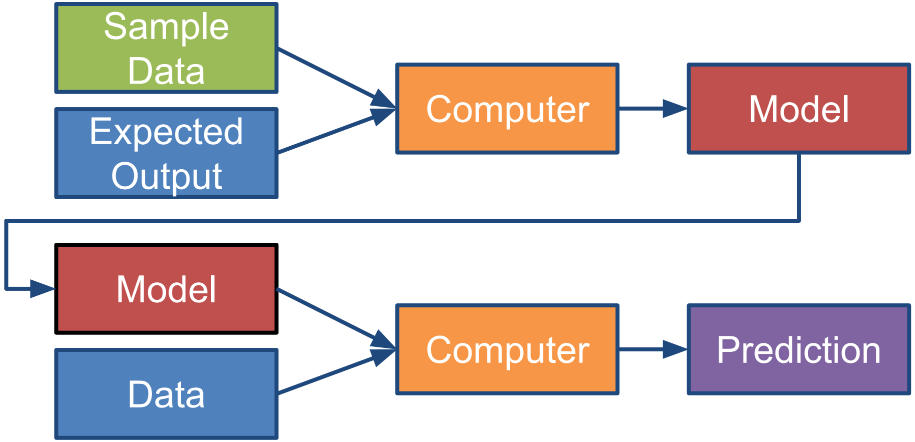




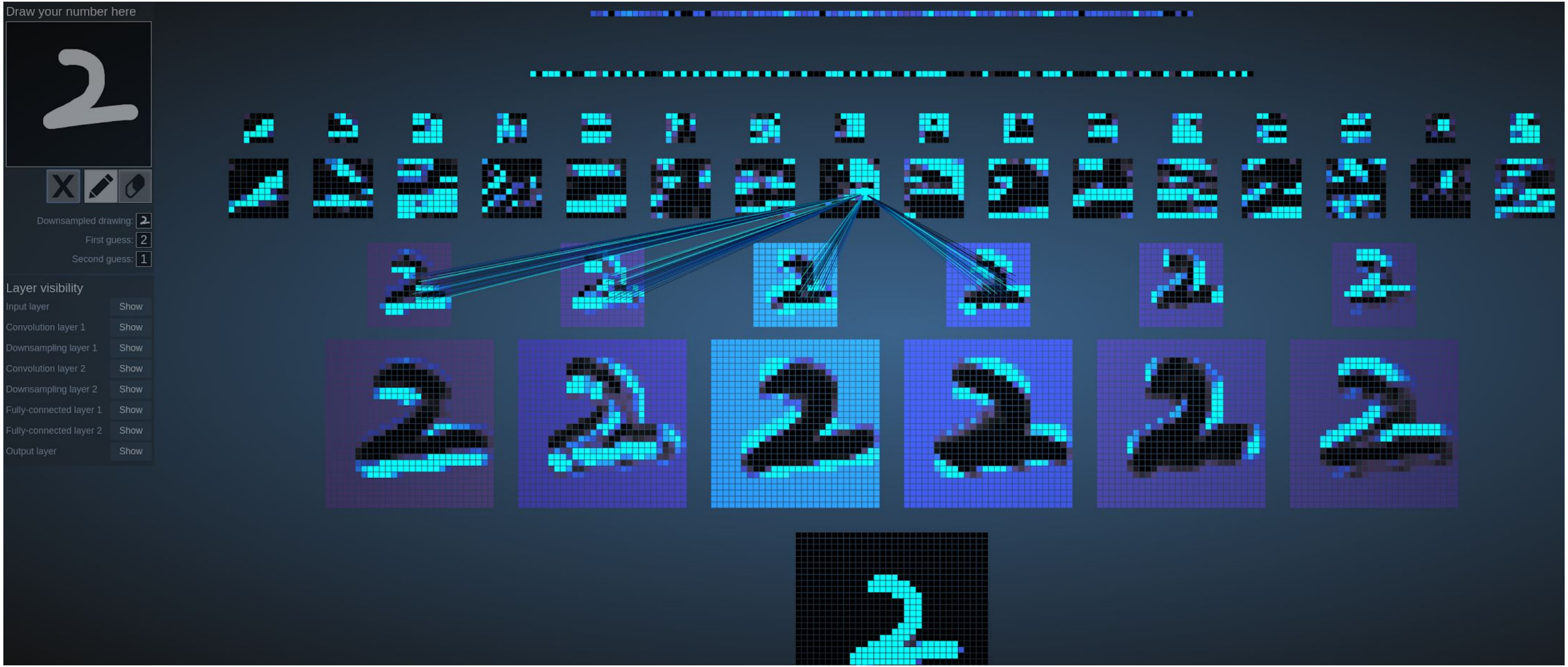
Traditional Modeling



Machine Learning (Supervised Learning)

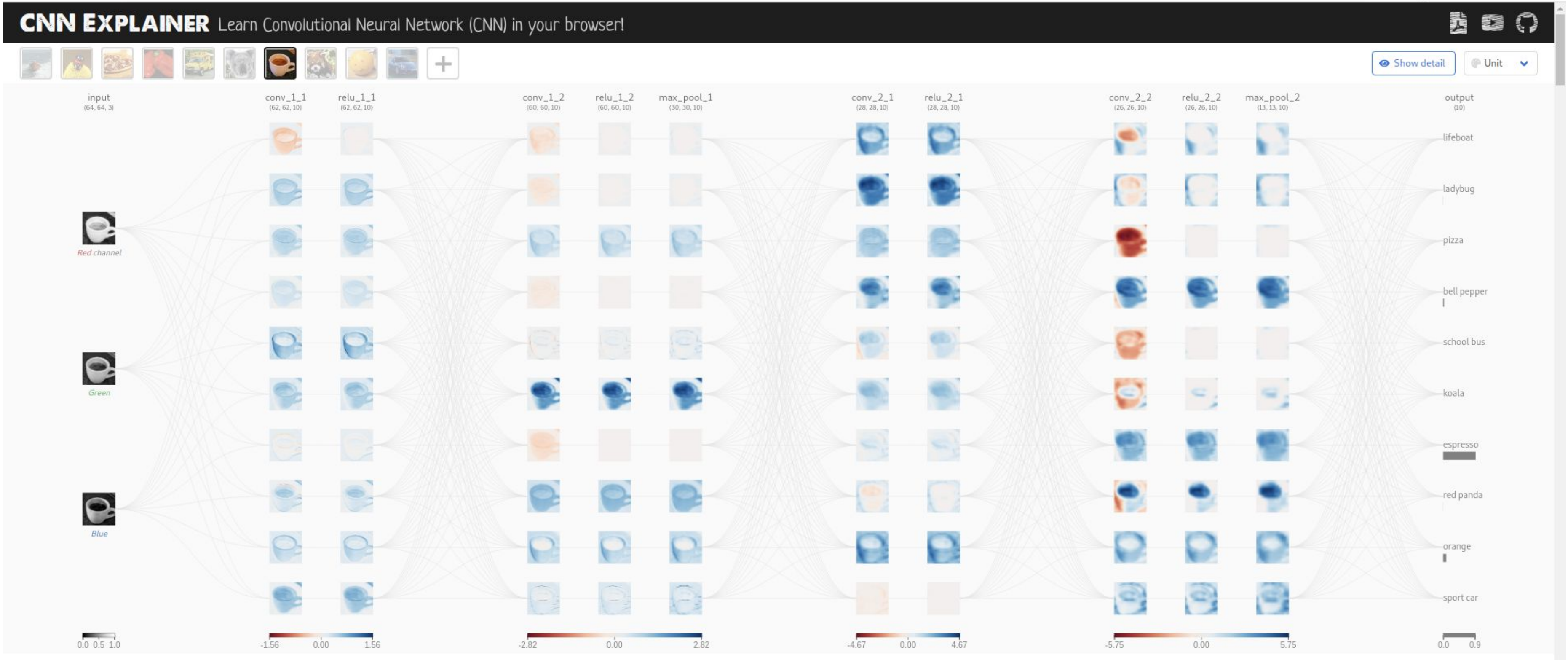


MNIST - CNN Visualization



(Image Credit: https://adamharley.com/nn_vis/cnn/3d.html)

CNN Explainer



(Image Credit: https://poloclub.github.io/cnn-explainer/)

Knowledge and Data Resources

ACES Website	https://hprc.tamu.edu/aces/
ACES Quick Start Guide	https://hprc.tamu.edu/kb/Quick-Start/ACES/
ACES Portal (ACCESS)	https://portal-aces.hprc.tamu.edu/pun/sys/dashboard
ACCESS Documentation	https://support.access-ci.org/
Texas A&M HPRC YouTube Channel	https://www.youtube.com/texasamhprc

help@hprc.tamu.edu

Repositories

Nvidia Deep Learning Examples	https://github.com/NVIDIA/DeepLearningExamples
Intel AI Models	https://github.com/intel/models
Graphcore Machine Learning Examples	https://github.com/graphcore/examples

Shared Datasets and Resources on ACES

- Datasets: ImageNet datasets for PyTorch and TensorFlow
  /scratch/data/pytorch-computer-vision-datasets
  /scratch/data/tensorflow-computer-vision-datasets

- NVIDIA AI models

  /scratch/data/nvidia-ai-models
- Intel AI models:

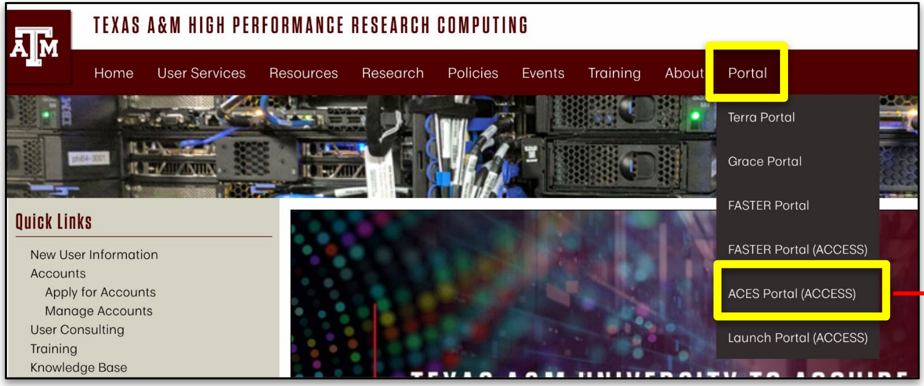
  /scratch/data/intel-ai-models
- Containers
  /scratch/data/pytorch-max-series-multi-node-multi-card-training.sif
  /scratch/data/containers/nvidia-containers

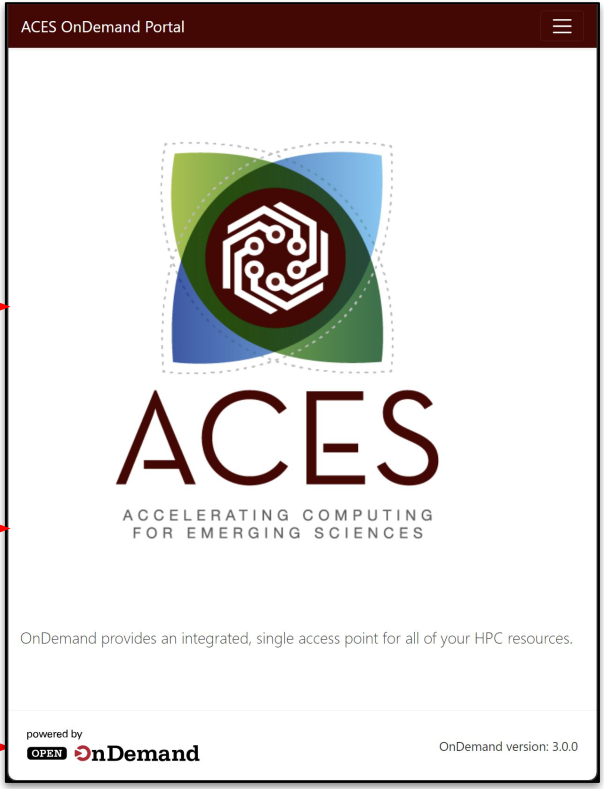
ACES Accelerators Here!

Component	Quantity	Description
Graphcore IPU (Colossus)	16	Colossus GC200 IPUs. Each IPU group hosted with a CPU server as a POD16 on a 100 GbE RoCE fabric
Graphcore IPU (BOW)	16	Bow IPUs. Each IPU group hosted with a CPU server as a POD16 on a 100 GbE RoCE fabric
NVIDIA H100	30	NVIDIA GPUs for HPC, DL Training
NVIDIA A30	4	NVIDIA GPUs for HPC, DL Training, AI Inference
Intel GPU Max 1100 (PVC)	120	Intel GPUs for HPC, DL Training, AI Inference

Accessing ACES

ACES Portal

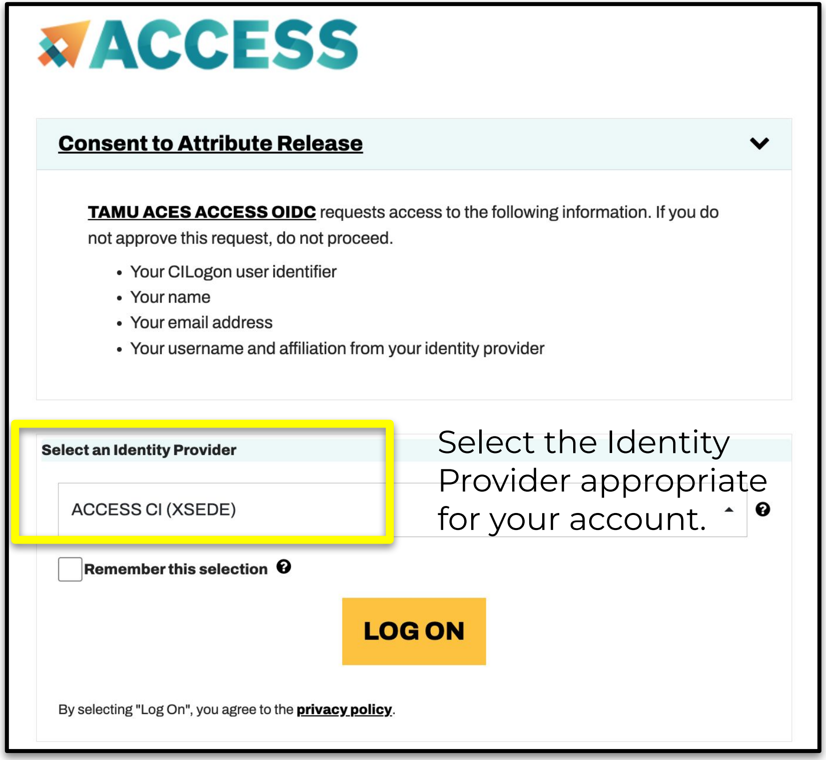




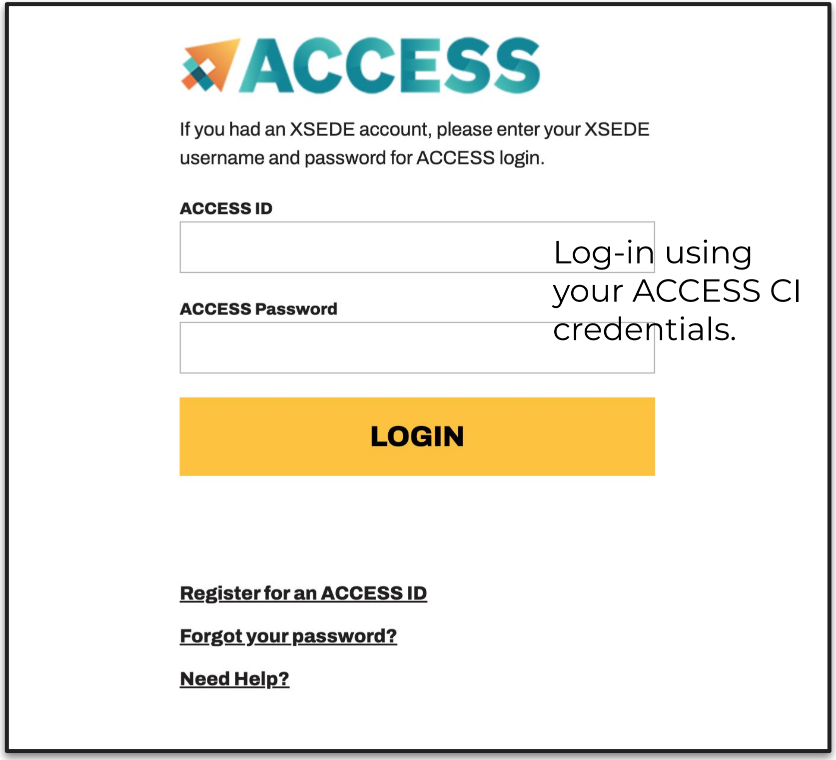
Log into the ACES portal
portal-aces.hprc.tamu.edu

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

Accessing via ACCESS

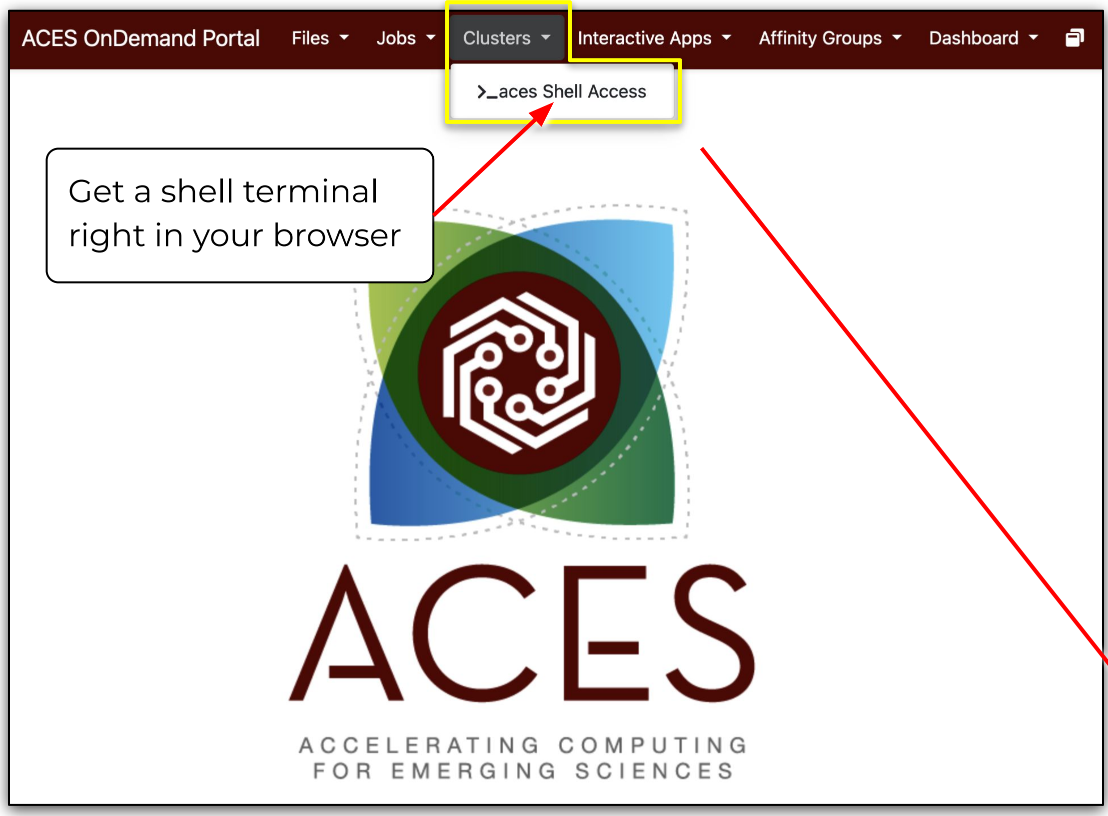


Select the Identity Provider appropriate for your account.



Log-in using your ACCESS CI credentials.

Shell Access via the Portal



Get a shell terminal right in your browser


```
********************************************************************************
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 motd 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 ~]$
```

Accelerator Status Check

- View the Nvidia GPU nodes and number of GPUs

```
$ pestat -p gpu -G
```

- View the Intel PVC nodes and number of GPUs

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

- Additional features on the PVC nodes

```
$ show_pvc_features
```

- IPU status will be introduced later

Copy Training Materials to Your Directory

- Navigate to your scratch directory

  `$ cd $SCRATCH`

- Files for this course are located at

  `/scratch/training/aces_wkshop_24`

  Make a copy in your personal scratch directory

  `$ cp -r /scratch/training/aces_wkshop_24 $SCRATCH`

- Enter this directory (your local copy)

  `$ cd $SCRATCH/aces_wkshop_24`

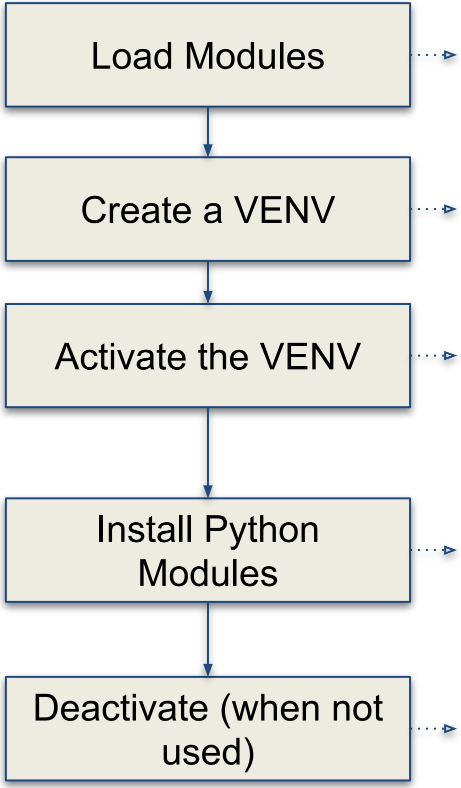
AI/ML on Nvidia GPUs

ACES

ACCELERATING COMPUTING
FOR EMERGING SCIENCES

NVIDIA®

Env Setup (option 1)




```
# clean up and load Anaconda
cd $SCRATCH
module purge
module load Anaconda3/2022.10

# create a Python virtual environment
conda create -n pytorch-env

# activate the virtual environment
source activate pytorch-env

# install required package to be used in the portal
conda install pytorch torchvision torchaudio \
pytorch-cuda=11.8 -c pytorch -c nvidia

# deactivate the virtual environment
# source deactivate
```

Use Shared Environment (option 2)

- A shared environment was created by TAMU HPRC on ACES cluster
- Path: */sw/hprc/sw/Anaconda3/2022.10/envs/pytorch-env*

Hands-on Session

- Navigate to the nvidia-gpu exercises directory

```
$ cd $SCRATCH/aces_wkshop_24/nvidia-gpu
```

- Open the python file (*main.py*) with your preferred editor (e.g. vim).
- Read the code in the file.
- Submit your job to Nvidia GPU queue.

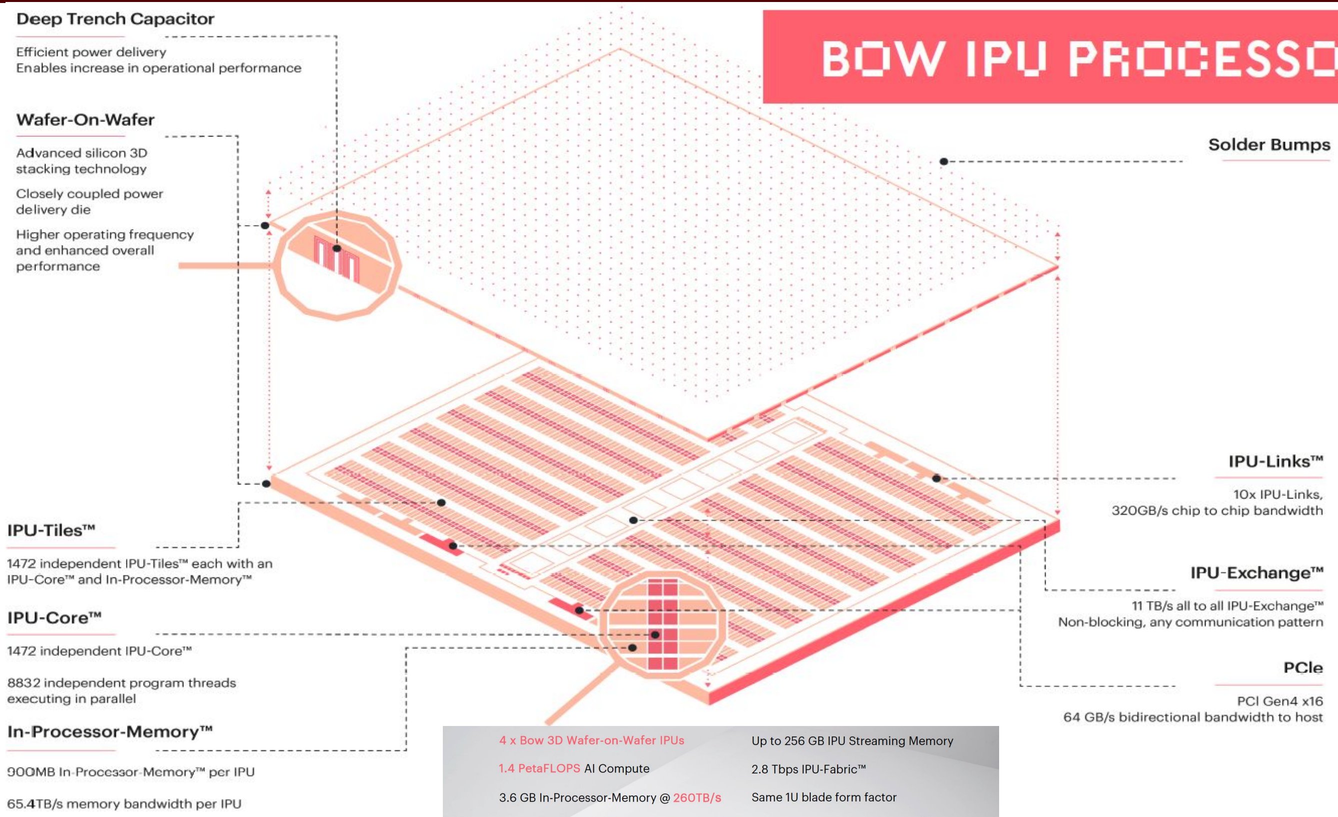
```
$ sbatch submit_job.sh
```

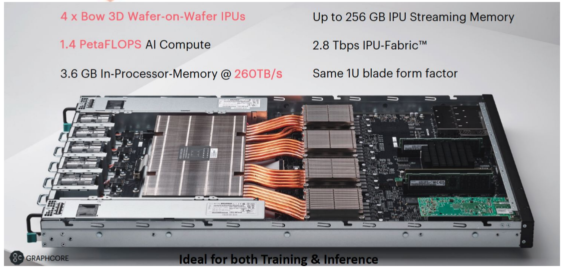
AI/ML on Graphcore IPUs

ACES

ACCELERATING COMPUTING FOR EMERGING SCIENCES

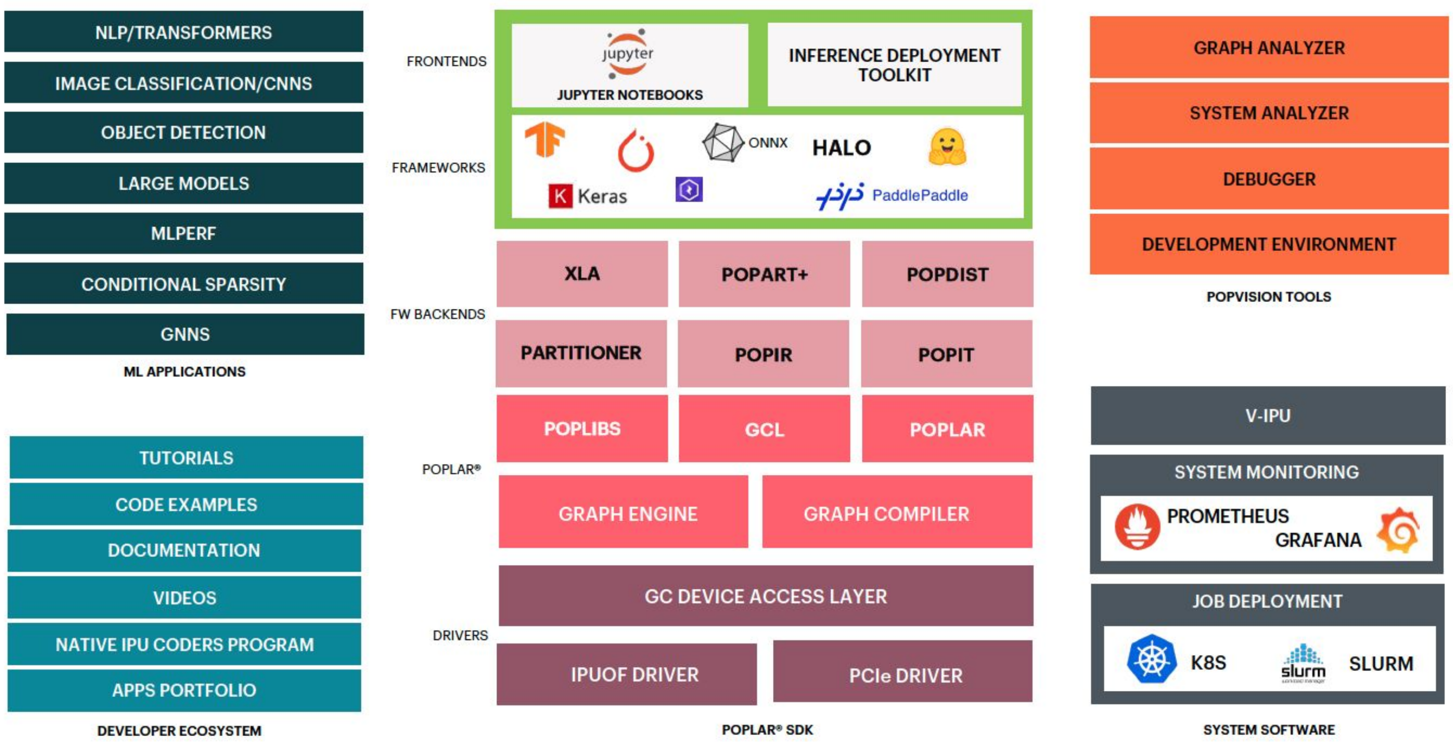
BOW IPU PROCESSOR





Source: Graphcore

Graphcore Software Stack



Source: Graphcore

Models on Graphcore GitHub

Vision	ResNet50, EfficientNet, DINO, MAE, Neural Image Fields, SWIN ((Shifted Windows Vision Transformers)), U-Net, ViT (Vision Transformer), YOLOv4, etc.
NLP	BERT, BLOOM-176B (BigScience Large Open-science Open-access Multilingual), GPT-2, GPT-3 2.7B, GPT-3 175B, GPT-J, etc.
Speech	Conformer, FastPitch, etc.
GNN	Cluster-GCN, GIN (Graph Isomorphism Network), NBFnet (Neural Bellman-Ford networks), SchNet, Spektral, TGN (Temporal Graph Networks), etc.
Multi-modal	CLIP, Frozen in time, MAGMA (Multimodal Augmentation of Generative Models through Adapter-based Finetuning), Mini DALL-E, etc.

Source: https://github.com/graphcore/examples

Training Materials

From the ACES login node, ssh into the poplar2 (BOW Pod16) IPU system:

```
ssh poplar2
```

Change to your scratch directory:

```
cd /localdata/$USER && mkdir ipu_labs && cd ipu_labs
```

Copy the example materials to your scratch directory:

```
git clone https://github.com/graphcore/examples.git
```

Copy the hands-on exercise materials to your scratch directory:

```
git clone https://github.com/happidence1/IPU-Training.git
```

Poplar Software Development Kit (SDK) setup

```
source
/opt/gc/poplar/poplar_sdk-ubuntu_20_04-3.3.0+1403-208993bbb7/poplar-ub
untu_20_04-3.3.0+7857-b67b751185/enable.sh

source
/opt/gc/poplar/poplar_sdk-ubuntu_20_04-3.3.0+1403-208993bbb7/popart-ub
untu_20_04-3.3.0+7857-b67b751185/enable.sh

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

Run a TensorFlow (TF) model on Bow IPU

TF Virtual Environment Setup

```
virtualenv -p python3 venv_tf2

source venv_tf2/bin/activate

python -m pip install
/opt/gc/poplar/poplar_sdk-ubuntu_20_04-3.3.0+1403-208993bbb7/
tensorflow-2.6.3+gc3.3.0+251582+08d96978c7f+intel_skylake512-
cp38-cp38-linux_x86_64.whl
```

Run a TensorFlow model on IPU

```
cd examples/tutorials/tutorials/tensorflow2/keras/completed_demos/

python completed_demo_ipu.py
```

- Deactivate the virtual environment after the model finishes running.

```
deactivate
```

Monitor IPU Usage - *gc-monitor*

- 4 partitions
- 16 IPUs
- Processes

```
watch -n 2 gc-monitor
```

```
Every 2.0s: gc-monitor                                                    poplar2: Thu Jul  6 10:33:31 2023

+---------------+------------------------------------------------------------------------------------+
|  gc-monitor   |                 Partition: p17 [active] has 16 reconfigurable IPUs                 |
+-------------+--------------------+--------+--------------+----------+-------+----+------+-------+
|    IPU-M    |       Serial       |IPU-M SW|Server version|  ICU FW  | Type  | ID | IPU# |Routing|
+-------------+--------------------+--------+--------------+----------+-------+----+------+-------+
|  10.5.5.1   | 0019.0002.8222521  | 2.6.0  |    1.11.0    |  2.5.9   | M2000 | 0  |  3   |  DNC  |
|  10.5.5.1   | 0019.0002.8222521  | 2.6.0  |    1.11.0    |  2.5.9   | M2000 | 1  |  2   |  DNC  |
|  10.5.5.1   | 0019.0001.8222521  | 2.6.0  |    1.11.0    |  2.5.9   | M2000 | 2  |  1   |  DNC  |
|  10.5.5.1   | 0019.0001.8222521  | 2.6.0  |    1.11.0    |  2.5.9   | M2000 | 3  |  0   |  DNC  |
+-------------+--------------------+--------+--------------+----------+-------+----+------+-------+
|  10.5.5.2   | 0021.0002.8222521  | 2.6.0  |    1.11.0    |  2.5.9   | M2000 | 4  |  3   |  DNC  |
|  10.5.5.2   | 0021.0002.8222521  | 2.6.0  |    1.11.0    |  2.5.9   | M2000 | 5  |  2   |  DNC  |
|  10.5.5.2   | 0021.0001.8222521  | 2.6.0  |    1.11.0    |  2.5.9   | M2000 | 6  |  1   |  DNC  |
|  10.5.5.2   | 0021.0001.8222521  | 2.6.0  |    1.11.0    |  2.5.9   | M2000 | 7  |  0   |  DNC  |
+-------------+--------------------+--------+--------------+----------+-------+----+------+-------+
|  10.5.5.3   | 0013.0002.8222521  | 2.6.0  |    1.11.0    |  2.5.9   | M2000 | 8  |  3   |  DNC  |
|  10.5.5.3   | 0013.0002.8222521  | 2.6.0  |    1.11.0    |  2.5.9   | M2000 | 9  |  2   |  DNC  |
|  10.5.5.3   | 0013.0001.8222521  | 2.6.0  |    1.11.0    |  2.5.9   | M2000 | 10 |  1   |  DNC  |
|  10.5.5.3   | 0013.0001.8222521  | 2.6.0  |    1.11.0    |  2.5.9   | M2000 | 11 |  0   |  DNC  |
+-------------+--------------------+--------+--------------+----------+-------+----+------+-------+
|  10.5.5.4   | 0016.0002.8222521  | 2.6.0  |    1.11.0    |  2.5.9   | M2000 | 12 |  3   |  DNC  |
|  10.5.5.4   | 0016.0002.8222521  | 2.6.0  |    1.11.0    |  2.5.9   | M2000 | 13 |  2   |  DNC  |
|  10.5.5.4   | 0016.0001.8222521  | 2.6.0  |    1.11.0    |  2.5.9   | M2000 | 14 |  1   |  DNC  |
|  10.5.5.4   | 0016.0001.8222521  | 2.6.0  |    1.11.0    |  2.5.9   | M2000 | 15 |  0   |  DNC  |
+-------------+--------------------+--------+--------------+----------+-------+----+------+-------+
+----------------------------------------------------------+-------------------------+----------------+
|           Attached processes in partition p17            |           IPU           |     Board      |
+--------+---------------------------+--------+------------+----+----------+--------+--------+-------+
|  PID   |          Command          |  Time  |    User    | ID |  Clock   |  Temp  |  Temp  | Power |
+--------+---------------------------+--------+------------+----+----------+--------+--------+-------+
| 902631 |          python           |  50s   | u.zh108696 | 0  | 1500MHz  | 23.5 C | 21.9 C | 90.3 W |
+--------+---------------------------+--------+------------+----+----------+--------+--------+-------+
```

Run a PyTorch (PopTorch) model on Bow IPU

PopTorch Virtual Environment Setup

```
cd /localdata/$USER/ipu_labs

virtualenv -p python3 poptorch_test

source poptorch_test/bin/activate

python -m pip install
/opt/gc/poplar/poplar_sdk-ubuntu_20_04-3.3.0+1403-208993bbb7/
poptorch-3.3.0+113432_960e9c294b_ubuntu_20_04-cp38-cp38-linux
_x86_64.whl
```

Run a PopTorch model on IPU

```
cd examples/tutorials/simple_applications/pytorch/mnist/

pip install -r requirements.txt

python mnist_poptorch.py
```

- Deactivate the virtual environment after the model finishes running.

```
deactivate
```

Monitor IPU Usage - *gc-monitor*

- 4 partitions
- 16 IPUs
- Processes
- IPU used
- Temperature
- Power

```
watch -n 2 gc-monitor
```

```
Every 2.0s: gc-monitor                                              poplar2: Thu Jul  6 10:55:55 2023

+---------------+------------------------------------------------------------------------------------+
|  gc-monitor   |              Partition: p17 [active] has 16 reconfigurable IPUs                    |
+-------------+--------------------+--------+------------+-----------+-------+----+-----+-------+
|    IPU-M    |       Serial       |IPU-M SW|Server version|  ICU FW  | Type  | ID | IPU# |Routing|
+-------------+--------------------+--------+------------+-----------+-------+----+-----+-------+
|  10.5.5.1   | 0019.0002.8222521  | 2.6.0  |    1.11.0    |  2.5.9   | M2000 | 0  |  3  |  DNC  |
|  10.5.5.1   | 0019.0002.8222521  | 2.6.0  |    1.11.0    |  2.5.9   | M2000 | 1  |  2  |  DNC  |
|  10.5.5.1   | 0019.0001.8222521  | 2.6.0  |    1.11.0    |  2.5.9   | M2000 | 2  |  1  |  DNC  |
|  10.5.5.1   | 0019.0001.8222521  | 2.6.0  |    1.11.0    |  2.5.9   | M2000 | 3  |  0  |  DNC  |
+-------------+--------------------+--------+------------+-----------+-------+----+-----+-------+
|  10.5.5.2   | 0021.0002.8222521  | 2.6.0  |    1.11.0    |  2.5.9   | M2000 | 4  |  3  |  DNC  |
|  10.5.5.2   | 0021.0002.8222521  | 2.6.0  |    1.11.0    |  2.5.9   | M2000 | 5  |  2  |  DNC  |
|  10.5.5.2   | 0021.0001.8222521  | 2.6.0  |    1.11.0    |  2.5.9   | M2000 | 6  |  1  |  DNC  |
|  10.5.5.2   | 0021.0001.8222521  | 2.6.0  |    1.11.0    |  2.5.9   | M2000 | 7  |  0  |  DNC  |
+-------------+--------------------+--------+------------+-----------+-------+----+-----+-------+
|  10.5.5.3   | 0013.0002.8222521  | 2.6.0  |    1.11.0    |  2.5.9   | M2000 | 8  |  3  |  DNC  |
|  10.5.5.3   | 0013.0002.8222521  | 2.6.0  |    1.11.0    |  2.5.9   | M2000 | 9  |  2  |  DNC  |
|  10.5.5.3   | 0013.0001.8222521  | 2.6.0  |    1.11.0    |  2.5.9   | M2000 | 10 |  1  |  DNC  |
|  10.5.5.3   | 0013.0001.8222521  | 2.6.0  |    1.11.0    |  2.5.9   | M2000 | 11 |  0  |  DNC  |
+-------------+--------------------+--------+------------+-----------+-------+----+-----+-------+
|  10.5.5.4   | 0016.0002.8222521  | 2.6.0  |    1.11.0    |  2.5.9   | M2000 | 12 |  3  |  DNC  |
|  10.5.5.4   | 0016.0002.8222521  | 2.6.0  |    1.11.0    |  2.5.9   | M2000 | 13 |  2  |  DNC  |
|  10.5.5.4   | 0016.0001.8222521  | 2.6.0  |    1.11.0    |  2.5.9   | M2000 | 14 |  1  |  DNC  |
|  10.5.5.4   | 0016.0001.8222521  | 2.6.0  |    1.11.0    |  2.5.9   | M2000 | 15 |  0  |  DNC  |
+-------------+--------------------+--------+------------+-----------+-------+----+-----+-------+

+--------------------------------------------------+-------------------------+----------------+
|          Attached processes in partition p17     |           IPU           |     Board      |
+--------+----------------------+-------+-----------+----+---------+--------+--------+-------+
|  PID   |       Command        | Time  |   User    | ID |  Clock  |  Temp  |  Temp  | Power |
+--------+----------------------+-------+-----------+----+---------+--------+--------+-------+
| 907530 |        python        |  17s  | u.zh108696| 0  | 1500MHz | 23.5 C | 22.0 C | 90.8 W |
+--------+----------------------+-------+-----------+----+---------+--------+--------+-------+
```

Hands-On Session

- Please access ACES and poplar2 now.
- Copy the tutorial materials to your scratch directory.
- Run the TensorFlow and PyTorch (PopTorch) example models on IPU

Porting TensorFlow Code to IPU

TensorFlow

Keras

1. Import the TensorFlow IPU module

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

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

2. Preparing the dataset

- Make sure the sizes of the datasets are divisible by the batch size

```
def make_divisible(number, divisor):
    return number - number % divisor
```

- Adjust dataset lengths

```
(x_train, y_train), (x_test, y_test) = load_data()
train_data_len = x_train.shape[0]
train_data_len = make_divisible(train_data_len, batch_size)
x_train, y_train = x_train[:train_data_len], y_train[:train_data_len]
test_data_len = x_test.shape[0]
test_data_len = make_divisible(test_data_len, batch_size)
x_test, y_test = x_test[:test_data_len], y_test[:test_data_len]
```

3. Add IPU configuration

To use the IPU, you must create an IPU session configuration:

```
ipu_config = ipu.config.IPUConfig()
ipu_config.auto_select_ipus = 1
ipu_config.configure_ipu_system()
```

A full list of configuration options is available in the API documentation.

4. Specify IPU strategy

```
strategy = ipu.ipu_strategy.IPUStrategy()
```

The `tf.distribute.Strategy` is an API to distribute training and inference across multiple devices. `IPUStrategy` is a subclass which targets a system with one or more IPUs attached.

5. Wrap the model within the IPU strategy scope

- Creating variables and Keras models within the scope of the `IPUStrategy` object will ensure that they are placed on the IPU.
- To do this, we create a `strategy.scope()` context manager and move all the model code inside it.

Hands-on Session

- Activate the TF virtual environment

```
cd /localdata/$USER/ipu_labs
source venv_tf2/bin/activate
```

- Change directory to Keras

```
cd IPU-Training/Keras
```

- Complete the **#Todo**s in the mnist-ipu-todo.py file.
- Run it in the **venv_tf2** virtual environment.

```
python mnist-ipu-todo.py
```

- After finishing the job, you can deactivate the virtual environment

```
deactivate
```

Porting PyTorch Code to Bow IPU

PopTorch

- PopTorch is a set of extensions for PyTorch released by Graphcore to enable PyTorch models to run on Graphcore's IPU hardware.

- PopTorch will use PopART to parallelise the model over the given number of IPUs. Additional parallelism can be expressed via a replication factor, which enables you to data-parallelise the model over more IPUs.

Training a model on IPU

- Import the packages

```
import torch
import poptorch
import torchvision
import torch.nn as nn
import matplotlib.pyplot as plt
from tqdm import tqdm
from sklearn.metrics import accuracy_score
```

Load the Data

- PyTorch: `torch.utils.data.DataLoader` class
- PopTorch extension: `poptorch.DataLoader` class,
  - Specialized for the way the underlying PopART framework handles batching of data.

Build the Model

```
class ClassificationModel(nn.Module):
    def __init__(self):
        super().__init__()
        self.conv1 = nn.Conv2d(1, 5, 3)
        self.pool = nn.MaxPool2d(2, 2)
        self.conv2 = nn.Conv2d(5, 12, 5)
        self.norm = nn.GroupNorm(3, 12)
        self.fc1 = nn.Linear(972, 100)
        self.relu = nn.ReLU()
        self.fc2 = nn.Linear(100, 10)
        self.log_softmax =
nn.LogSoftmax(dim=1)
        self.loss = nn.NLLLoss()
```

```
def forward(self, x, labels=None):
        x =
self.pool(self.relu(self.conv1(x)))
        x =
self.norm(self.relu(self.conv2(x)))
        x = torch.flatten(x, start_dim=1)
        x = self.relu(self.fc1(x))
        x = self.log_softmax(self.fc2(x))
        # The model is responsible for the
calculation of the loss when using an IPU.
We do it this way:
        if self.training:
            return x, self.loss(x, labels)
        return x


model = ClassificationModel()
model.train()
```

Prepare training for IPUs

The compilation and execution on the IPU can be controlled using `poptorch.Options`. These options are used by PopTorch's wrappers such as `poptorch.DataLoader` and `poptorch.trainingModel`.

```
opts = poptorch.Options()
train_dataloader = poptorch.DataLoader(
    opts, train_dataset, batch_size=16, shuffle=True, num_workers=20
)
```

Train the model

```
optimizer = poptorch.optim.SGD(model.parameters(), lr=0.001, momentum=0.9)

poptorch_model = poptorch.trainingModel(model, options=opts,
optimizer=optimizer)

epochs = 30
for epoch in tqdm(range(epochs), desc="epochs"):
    total_loss = 0.0
    for data, labels in tqdm(train_dataloader, desc="batches", leave=False):
        output, loss = poptorch_model(data, labels)
        total_loss += loss

poptorch_model.detachFromDevice()

torch.save(model.state_dict(), "classifier.pth")
```

Evaluate the Model

```
model = model.eval()

poptorch_model_inf = poptorch.inferenceModel(model, options=opts)

test_dataloader = poptorch.DataLoader(opts, test_dataset, batch_size=32,
     num_workers=10)

predictions, labels = [], []
for data, label in test_dataloader:
    predictions += poptorch_model_inf(data).data.max(dim=1).indices
    labels += label

poptorch_model_inf.detachFromDevice()

print(f"Eval accuracy: {100 * accuracy_score(labels, predictions):.2f}%")
```

Hands-on Session

- Activate the Poptorch virtual environment

```
cd /localdata/$USER/ipu_labs
source poptorch_test/bin/activate
```

- Change directory to PyTorch

```
cd IPU-Training/PyTorch
```

- Complete the **#Todos** in the `fashion-mnist-pytorch-ipu-todo.py` file.
- Run it in the **poptorch_test** virtual environment.

```
pip install -r requirements.txt
python fashion-mnist-pytorch-ipu-todo.py
```

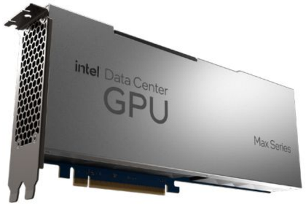
- After finishing the job, you can deactivate the virtual environment

```
deactivate
```

AI/ML on Intel PVC GPUs

ACES

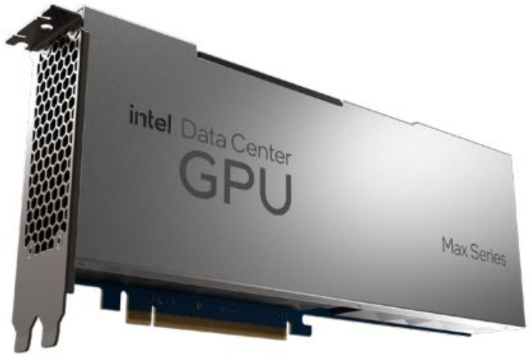
ACCELERATING COMPUTING FOR EMERGING SCIENCES



Intel Data Center GPU Max Series PCIE Card

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)

Using PVCs on ACES

PyTorch TensorFlow

Environment Setup for PyTorch Models

Use Intel AI Analytics Toolkit

```
# load all the necessary modules
module purge
module load intel/AIKit/2023.2.0
module load intel/2023.07

ENV_NAME=aikit-pt-gpu-clone

# If it doesn't exist, create the environment
if ! conda env list | grep -q "$ENV_NAME"; then

    conda create -n $ENV_NAME --clone aikit-pt-gpu
fi

# activate the conda environment
source activate $ENV_NAME
```

in pt_demo.slurm

Environment Setup for PyTorch Models

Use Python Virtual Environment (Alternative for reference)

```
# Change to pytorch directory
cd $SCRATCH/aces_pvc_course_24s/pytorch

# Load modules
module load WebProxy
module load intel/2023.03
module load Python/3.10.8

# Create and activate a Python virtual environment
python -m venv pt-pvc-labs
source pt-pvc-labs/bin/activate
```

Please do not type

Environment Setup for PyTorch Models

Use Python Virtual Environment (Alternative for reference)

```
# Install torch, torchvision and oneccl_bindings_for_pytorch
python -m pip install torch==1.13.0a0+git6c9b55e
torchvision==0.14.1a0 intel_extension_for_pytorch==1.13.120+xpu -f
https://developer.intel.com/ipex-whl-stable-xpu

python -m pip install oneccl_bind_pt==1.13.200+gpu -f
https://developer.intel.com/ipex-whl-stable-xpu

# Install tensorboard
python -m pip install tensorboard
```

Please do not type

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`

Environment Setup for TensorFlow Models

Using the Intel AI Analytics Toolkit

```
# load all the necessary modules
module purge
module load WebProxy
module load intel/2023.07
module load intel/AIKit/2023.1.0

ENV_NAME=aikit-tf-gpu-clone

# If it doesn't exist, create the environment
if ! conda env list | grep -q "$ENV_NAME"; then

    conda create -n $ENV_NAME --clone aikit-tf-gpu
fi

# activate the conda environment
source activate $ENV_NAME
```

in tf_demo.slurm

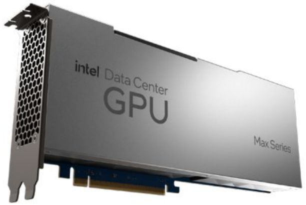
Run Tensorflow ResNet50 Model

- 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
```

To Continue at PEARC24

ACES
ACCELERATING COMPUTING
FOR EMERGING SCIENCES



Intel Data Center GPU Max Series PCIE Card

intel

Texas A&M at PEARC24

Talk/Event	Date/Time	Room
Tutorial: Hands-on exercises on the Intel Data Center GPU Max 1100 (PVC-GPU) for AI/ML and Molecular	Mon, July 22, 2024 9:00 AM-12:30 PM ET	Room 553B
Seventh Workshop on Strategies for Enhancing HPC Education and Training (SEHET24)	Mon, July 22, 2024 9:00 AM-12:30 PM ET	Room 557
Workshop: Providing cutting-edge computing testbeds to the science and engineering community	Mon, July 22, 2024 1:30 PM-5:00 PM ET	Room 554A
Workshop: Engaging Secondary Students in Computing: K12 Outreach	Mon, July 22, 2024 1:30 PM-5:00 PM ET	Room 553A
Cultivating Cyberinfrastructure Careers through Student Engagement at Texas A&M University High Performance Research Computing	Tue, July 23, 2024 11:00 AM-11:25 AM ET	Junior Ballroom
Insight Gained from Migrating a Machine Learning Model to Intelligence Processing Units	Tue, July 23, 2024 11:00 AM-11:25 AM ET	Room 551 A&B
BOF 4: What's in it for me? How can we truly democratize the research computing and data community?	Tue, July 23, 2024 1:30 PM-2:30 PM ET	Room 551 A&B

Texas A&M at PEARC24

Talk/Event	Date/Time	Room
BRICCs: Building Pathways to Research Cyberinfrastructure at Under Resourced Institutions	Tue, July 23, 2024 3:25 PM-3:50 PM ET	Junior Ballroom
Memory Bandwidth Performance across Accelerators	Tue, July 23, 2024 3:25 PM-3:50 PM ET	Ballroom B
Container Adoption in Campus High Performance Computing	Wed, July 24, 2024 11:00 AM-11:25 AM ET	Ballroom B
Engaging Secondary Students in Computing and Cybersecurity	Wed, July 24, 2024 3:15 PM-3:30 PM ET	Room 557
Exploring the Viability of Composable Architectures to Overcome Memory Limitations in High Performance Computing Workflows	Wed, July 24, 2024 3:45 PM-4:00 PM ET	Room 553 A&B
Performance of Molecular Dynamics Acceleration Strategies on Composable Cyberinfrastructure	Wed, July 24, 2024 4:15 PM-4:30 PM ET	Room 551 A&B
BOF 17: Fantastic ACCESS Cyberinfrastructure Resources and Where to Find Them	Wed, July 24, 2024 4:45 PM-5:45 PM ET	Room 553 A&B
BOF 18: Recipes to build successful cross-institutional collaborative computing	Wed, July 24, 2024 4:45 PM-5:45 PM ET	Junior Ballroom

High Performance
Research Computing
DIVISION OF RESEARCH

Thank you

- We gratefully acknowledge support from National Science Foundation awards #2112356 (ACES).
- Please visit our talks and BoFs at PEARC24
- Helpdesk: help@hprc.tamu.edu