AI/ML Frameworks and Advanced Computing Resources to Accelerate Research at Texas A&M’s High Performance Research Computing (HPRC) Facility

CESG Seminar
February 4, 2022

High Performance Research Computing
DIVISION OF RESEARCH
High Performance Research Computing

Our Mission:

- Provide **computing** resources
- Provide consulting, technical guidance, and training to support users of these resources.
- Collaborate on computational and data-enabled research.

Credit: towardsdatascience.com
HPRC Services

- **Free of charge** to all faculty, postdocs, research staff, students and external collaborators
- Computing cycles for research and university course purposes
- **Application is required for access**
- User Services
  - Helpdesk: New user start-up assistance and general support
  - Training: Short Courses, Workshops, & YouTube videos
  - Advanced Support: Software and research consulting
    - Expertise in many science and engineering research domains
- Access to state and national advanced computing resources
# High Performance Research Computing Clusters

## Grace
- **Total Nodes (Cores):** 925 (44,656)
- **General Nodes:** 48 cores, 384GB
- **Features:** GPUs (A100, RTX 6000, T4), Large Memory Nodes
- **Interconnect:** HDR100 InfiniBand
- **Global Disk (raw):** 8.9 PB

## Terra
- **Total Nodes (Cores):** 307 (8,512)
- **General Nodes:** 28 cores, 64GB
- **Features:** GPUs (K80, V100), KNL
- **Interconnect:** Omni-Path
- **Global Disk (raw):** 7.4 PB

## ViDaL
- **Total Nodes (Cores):** 24 (1,120)
- **General Nodes:** 40 cores, 192 GB
- **Features:** Compliant Computing GPUs (V100), Large Memory Nodes
- **Interconnect:** 40Gb Ethernet
- **Global Disk (raw):** 2 PB

[https://hprc.tamu.edu/resources](https://hprc.tamu.edu/resources)
## HPRC Account Allocations

<table>
<thead>
<tr>
<th>Allocation Type</th>
<th>Who can apply?</th>
<th>Minimum SUs per Allocation per Machine</th>
<th>Maximum SUs per Machine</th>
<th>Maximum Total SUs per Machine</th>
<th>Maximum Number of Allocations per Machine</th>
<th>Allowed to spend more than allocation?</th>
<th>Reviewed and approved by</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic</td>
<td>Faculty, Post-Docs*, Research Associates, Research Scientists, Qualified Staff, Students*, Visiting Scholars/Students*</td>
<td>5,000</td>
<td>5,000</td>
<td>5,000</td>
<td>1</td>
<td>No</td>
<td>HPRC Staff</td>
</tr>
<tr>
<td>Startup</td>
<td>Faculty, Research Associates, Research Scientists, Qualified Staff</td>
<td>5,000</td>
<td>200,000</td>
<td>400,000</td>
<td>2</td>
<td>No</td>
<td>HPRC Director</td>
</tr>
<tr>
<td>Research (Terra)</td>
<td>Faculty, Research Scientists, Qualified Staff</td>
<td>300,000</td>
<td>5,000,000</td>
<td>5,000,000</td>
<td>Determined by HPRC-RAC</td>
<td>No</td>
<td>HPRC-RAC</td>
</tr>
<tr>
<td>Research (Grace)</td>
<td>Faculty, Research Scientists, Qualified Staff</td>
<td>300,000</td>
<td>10,000,000</td>
<td>10,000,000</td>
<td>Determined by HPRC-RAC</td>
<td>No</td>
<td>HPRC-RAC</td>
</tr>
</tbody>
</table>

Students & Postdoctoral researchers can apply for a Basic allocation.

PIs can apply for a Startup or Research allocation and sub-allocate SUs to their researchers.

[https://hprc.tamu.edu/policies/allocations.html](https://hprc.tamu.edu/policies/allocations.html)
HPRC Account: PI Eligibility

Only active faculty members and permanent research staff (subject to HPRC-RAC Chair review and approval) of Texas A&M System Members headquartered in Brazos County can serve as a PI.

Adjunct and Visiting professors can use HPRC resources as part of a sponsoring PI's group

Note that:

- A PI can have more than one allocation
- A researcher (student) can work on more than one project and with more than one PI

https://hprc.tamu.edu/policies/allocations.html
HPRC Training Short Courses
https://hprc.tamu.edu/training

**Primers:**
- Linux
- HPRC Clusters
- Data Management
- SLURM
- Jupyter Notebook

**Short Courses:**
- Python
- Scientific Python
- PyTorch
- TensorFlow
- MATLAB
- Scientific ML
- Julia
- CUDA
- Drug Docking
- Quantum Chemistry
- and more...

**Technology Lab:**
- Using AI Frameworks in Jupyter Notebook

**Short Courses:**
- NGS Analysis
- NGS Metagenomics
- NGS RADSeq/GBS
- NGS Assembly
- HPRC Galaxy
- Linux
- R
- Perl
- Fortran
- OpenMP
- MPI
Upcoming HPRC Short Courses

**Introduction to Python**

**Instructor:** Richard Lawrence  
**Time:** Friday, February 18, 10:00AM-12:30PM  
**Location:** Blocker 220 and online using Zoom  
**Description:** Covers basic topics in programming using Python. Topics include variables, data types, control statements, functions, I/O, modules, interactive execution of python statements, python scripts, dictionaries, sorting, and regular expressions.  
**Prerequisites:** HPRC account

**Introduction to Scientific Python**

**Instructor:** Zhenhua He  
**Time:** Friday, February 18, 1:30PM-4:00PM  
**Location:** Blocker 220 and online using Zoom  
**Description:** This short course covers several topics and packages for scientific programming in Python.  
**Prerequisites:** Basic Python skills, HPRC account
Upcoming HPRC training
Technology Lab: Using AI Frameworks in Jupyter Notebook
Instructor: Zhenhua He
Time: Early March
Registration available soon: https://hprc.tamu.edu/training

To help a new user start with their machine learning projects on HPRC systems

- **Lab I. JupyterLab (30 mins)**
  We will set up a Python virtual environment and run JupyterLab on the HPRC Terra Portal.

- **Lab II. Data Exploration (30 mins)**
  We will go through some examples with two popular Python libraries: Pandas and Matplotlib for data exploration.

- **Q&A (5 mins/lab)**

- **Lab III. Machine Learning (30 mins)**
  We will learn to use scikit-learn library for linear regression and classification applications.

- **Lab IV. Deep Learning (30 minutes)**
  We will learn how to use Keras to create and train a simple image classification model with deep neural network (DNN).
Open OnDemand (OOD) Portal is an advanced web-based graphical interface for HPC users.

Interactive Apps: launch a software window right in your browser.

HPRC Portal
https://portal.hprc.tamu.edu

HPRC Portal
YouTube tutorials
JupyterLab Portal Support
JupyterLab Portal Support

- HPRC Portal
- shell access
- Create a virtual environment
- git clone the Jupyter notebooks
- Open the JupyterLab in interactive apps
# clean up and load Anaconda
\texttt{cd}\ $\texttt{SCRATCH}$
module purge
module load Python/3.7.4-GCCcore-8.3.0

# create a Python virtual environment
\texttt{python}\ -m\ \texttt{venv}\ \texttt{mylab}

# activate the virtual environment
source mylab/bin/activate

# install required package to be used in the portal
\texttt{pip}\ install\ --upgrade\ pip\ setuptools
\texttt{pip}\ install\ jupyterlab\ torch\ torchvision\ tensorboard
\texttt{pip}\ install\ pandas\ scikit-plot\ tqdm\ seaborn

# deactivate the virtual environment
# deactivate
HPRC Portal Dashboard

https://portal.hprc.tamu.edu
Containerized Jupyter Notebooks

NVIDIA TAO
NVIDIA Train, Adapt, and Optimize (TAO) is an AI-model adaptation platform that reduces the time to produce highly accurate models from hours to months.

Type of environment
Containers (Singularity)
Select the type of environment in which Jupyter is installed.

Path to singularity image file
/scratch/data/Singularity/images/tensorflow_2.4.1-gpu-jupyter.sif
Enter the path to a singularity image file containing the Jupyter app.

Node type
GPU
Choose "GPU" if the notebook needs to run on an Nvidia GPU node.
HPRC Portal - Interactive Apps

**BIO**
- Beauti
- DIYABC
- FigTree
- IGV
- JBrowse
- Krait
- Mauve
- Structure
- Tracer
- CRISPR-Local
- Gap5

**GUI**
- ANSYS Workbench
- Abaqus/CAE
- LS-PREPOST
- MATLAB
- ParaView
- VNC

**Servers**
- Jupyter Notebook
- JupyterLab
- RStudio
- Spark-Jupyter Notebook

**Imaging**
- AFNI
- Chimera
- Coot
- Diffusion Toolkit & TrackVis
- FSL
- Fiji
- ICY
- ImageJ
- Vaa3D
- cisTEM
# Available Software Modules

**SOFTWARE MODULES ON THE TERRA CLUSTER**

Last Updated: Mon Oct 12 00:00:02 CDT

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPflow</td>
<td>GPflow is a package for building Gaussian process models in python.</td>
</tr>
<tr>
<td>Horovod</td>
<td>Horovod is a distributed training framework for TensorFlow. URL: <a href="https://horovod.org/">https://horovod.org/</a></td>
</tr>
<tr>
<td>Keras</td>
<td>Keras is a minimalist, highly modular neural networks library, written of either TensorFlow or Theano. URL: <a href="https://keras.io/">https://keras.io/</a></td>
</tr>
<tr>
<td>segmentation-models</td>
<td>Python library with Neural Networks for Image Segmentation base. URL: <a href="https://github.com/qubvel/segmentation_models">https://github.com/qubvel/segmentation_models</a></td>
</tr>
<tr>
<td>TensorFlow</td>
<td>An open-source software library for Machine Intelligence URL: <a href="http://tensorflow.org/">http://tensorflow.org/</a></td>
</tr>
</tbody>
</table>

Showing 1 to 5 of 5 entries (filtered from 1,636 total entries)

[https://hprc.tamu.edu/software](https://hprc.tamu.edu/software)
Available Software Modules

https://hprc.tamu.edu/wiki/SW:Modules

mla command to quickly search for installed software:

mouse@terra2 ~]$ ml a scikit-learn
Using /home/mouse/module.avail.terra
scikit-learn/
cikit-learn/0.18.1-intel-2017A-Python-2.7.12
scikit-learn/0.19.1-foss-2017b-Python-2.7.14
scikit-learn/0.19.1-foss-2017b-Python-3.6.3
scikit-learn/0.19.1-foss-2018a-Python-3.6.4
scikit-learn/0.19.1-fosscuda-2017b-Python-3.6.3
......
cikit-learn/0.21.3-fosscuda-2019b-Python-3.7.4
cikit-learn/0.21.3-intel-2019b-Python-3.7.4
cikit-learn/0.22.1-intel-2019b-Python-3.7.4
cikit-learn/0.23.1-foss-2020a-Python-3.8.2
cikit-learn/0.23.1-fosscuda-2020a-Python-3.8.2
cikit-learn/0.23.1-intel-2020a-Python-3.8.2
cikit-learn/0.23.1-intelcuda-2020a-Python-3.8.2
cikit-learn/0.23.2-foss-2020b
cikit-learn/0.23.2-intel-2020b
cikit-learn/0.23.2-intelcuda-2020b

Python
Matlab
Keras
PyTorch
scikit-learn
Pandas
NumPy
Matplotlib
Julia
......
Compilers: C++, Fortran, Intel OneAPI, GNU, ...
CUDA, OpenCL
OpenMPI, IntelMPI
......
Advanced Support Program

Collaborations on computational research projects.

HPRC analysts can contribute expertise in:

- Software development for research workflows
- Developing GUIs and apps for research projects
- Porting applications to HPC clusters
- Code development, optimizing and analysis on serial and parallel platforms
- Leveraging mathematical libraries
- Workflow automation in scientific processes

Please send us an e-mail: help@hprc.tamu.edu

ASP is supported in part by NSF award #1925764, CC* Team: SWEETER -- SouthWest Expertise in Expanding, Training, Education and Research and NSF award #2112356, Category II: ACES - Accelerating Computing for Emerging Sciences
NSF MRI FASTER
Fostering Accelerated Scientific Transformations, Education, and Research

- **Composable** software-hardware approach
- 184-Intel Ice Lake nodes (11,520-core) with InfiniBand. (64-core, 256GB memory, and 3.84TB NVMe disk per node)
- **NVIDIA GPUs**: 200x T4, 40x A100, 10x A10, 4x A30, and 8x A40 GPUs
- Each node can compose up to 20 GPUs.

This project is supported by NSF award #2019129
High Performance Computing (HPC) Architecture Comparison

**Legacy HPC**
- Built on Converged HW
- Static Hardware Design
- Fixed GPUs/Accelerators
- Fixed Memory
- Legacy Storage: SATA and SAS

**Modern HPC**
- Built on Disaggregated HW
- Composable Hardware Platform
- Composable GPUs/Accelerators
- Composable Memory - Optane
- Modern Storage: NVMe-oF

Modern HPC Platforms Support Composable GPUs/Accelerators and Memory
ACES - Accelerating Computing for Emerging Sciences

ACES is an innovative advanced computational prototype to be developed by Texas A&M University partnering with TACC and UIUC.

September 23, 2021

As Moore’s law slows, HPC developers are increasingly looking for speed gains in specialized code and specialized hardware – but this specialization, in turn, can make testing and deploying code trickier than ever. Now, researchers from Texas A&M University, the University of Illinois at Urbana-Champaign and the University of Texas at Austin have teamed, with NSF funding, to build a $5 million prototype supercomputer (“ACES”) with a dynamically configurable smorgåsbord of hardware, aiming to support developers as hardware needs grow ever more diverse.

ACES (short for “Accelerating Computing for Emerging Sciences”) is presented as an “innovative composability hardware platform.” ACES will leverage a PCIe-based composable framework from Liquid to offer access to Intel’s high-bandwidth memory Sapphire Rapids processors and more than 20 accelerators: Intel FPGAs; NEC Vector Engines; NextSilicon co-processors; Graphcore IPUs (Intelligence Processing Units); and Intel’s forthcoming Ponte Vecchio GPUs. All this hardware will be coupled with Intel Optane memory and DDN Lustre Storage and connected with Mellanox NDR 400Gbps networking.

This project is supported by NSF award #2112356

"ACES will enable applications and workflows to dynamically integrate the different accelerators, memory, and in-network computing protocols to glean new insights by rapidly processing large volumes of data,” the NSF grant reads, “and provide researchers with a unique platform to produce complex hybrid programming models that effectively supports calculations that were not feasible before."

## ACES System Description

<table>
<thead>
<tr>
<th>Component</th>
<th>Quantity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allocatable resources</td>
<td></td>
<td>Total cores: 11,520</td>
</tr>
<tr>
<td>CPU-centric computing with variable memory requirements</td>
<td>120 nodes (11,520 cores)</td>
<td>Dual Intel Sapphire Rapids 2.1 GHz 48 core processors with HBM2e memory 96 cores per node, 512 GB memory, 1.6 TB NVMe storage (PCIe 5.0), NVIDIA Mellanox NDR 200 Gbps Infiniband</td>
</tr>
<tr>
<td>Composable infrastructure</td>
<td>120 nodes</td>
<td>Dynamically reconfigurable infrastructure that allows up to 20 PCIe cards (GPU, FPGA, VE, etc.) per compute node</td>
</tr>
<tr>
<td>Data transfer nodes</td>
<td>2 nodes</td>
<td>Same as compute nodes, 100 Gbps network adapter</td>
</tr>
</tbody>
</table>
# ACES - Accelerating Computing for Emerging Sciences
(To be deployed in 2022)

<table>
<thead>
<tr>
<th>Component</th>
<th>Quantity*</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Graphcore IPU</strong></td>
<td>16</td>
<td>16 IPUs direct-attached to a server</td>
</tr>
<tr>
<td><strong>Intel Agilex FPGA</strong></td>
<td>20</td>
<td>Agilex FPGA with a broad hierarchy of memory including DDR5, HBM2e and Optane Persistent Memory</td>
</tr>
<tr>
<td><strong>NextSilicon coprocessor</strong></td>
<td>20</td>
<td>Reconfigurable accelerator with an optimizer continuously evaluating application behavior.</td>
</tr>
<tr>
<td><strong>NEC Vector Engine</strong></td>
<td>24</td>
<td>Vector computing card with 8 cores and HBM2 memory</td>
</tr>
<tr>
<td><strong>Intel Ponte Vecchio GPU</strong></td>
<td>100</td>
<td>Intel GPU for HPC, DL Training, AI Inference</td>
</tr>
<tr>
<td><strong>Liqid Intel Optane PCIe SSDs</strong></td>
<td>6</td>
<td>3 TB PCIe SSD cards addressable as memory using Intel Memory Drive Technology</td>
</tr>
</tbody>
</table>

*Estimated quantities
Graphcore IPUs (Intelligence Processing Unit)

https://www.graphcore.ai/posts/accelerating-resnet50-training-on-the-ipu-behind-our-mlperf-benchmark
2. IMPORT THE TENSORFLOW IPU MODULE

First, we import the TensorFlow IPU module.

Add the import statement in Listing 2.1 to the beginning of your script.

Listing 2.1 Importing ipu Python module

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

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

3. IPU CONFIG

To use the IPU, you must create an IPU session configuration in the main process. A minimum configuration is in Listing 3.1.

Listing 3.1 Example of a minimum configuration

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

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

4. MODEL
Specialized Training

INTEL DEVELOPER TOOLS TRAINING
INTEL AI ANALYTICS TOOLKIT

Texas A&M University
January 21, 2022, 1:30 p.m. - 4:00 p.m. CST

Texas A&M High Performance Research Computing is inviting you to an online workshop to get introduced to Intel AI software and the performance benefits achieved from using the Intel optimizations. This workshop will be presented by Intel engineers. Participants will receive a certificate of completion from Intel.

Agenda

The workshop will cover Intel optimizations implemented on top of stock versions of data science libraries like NumPy, SciPy, Scikit Learn, and DL frameworks like Tensorflow and Pytorch. Hands-on exercises will be followed to showcase how to get started using Intel AI software and the performance benefits achieved from using Intel optimizations.

- Lecture - What is oneAPI - AI Analytics Toolkit - 30 min
- Intel Distribution for Python (IDP)
  - Skill Level - High level understanding of some data science Python libraries, Python beginner level
  - Overview of optimizations inside Python - 5 min
  - Exercise complete with instructor - 20 min
  - Exercise URL - https://github.com/motlubasevo/numpy-tasks
- Individual time to complete exercise, Q&A - 5 min
- Expected Outcome be able to see the performance benefit of using IDP libraries over stock Python libraries like NumPy, SciPy etc.
- Intel Extensions for Scikit Learn
  - Skill Level - High level understanding of Scikit Learn library, Python beginner level
  - Overview of optimizations inside Scikit Learn - 5 min
  - Exercise complete with instructor - 20 min.
- Individual time to complete exercise, Q&A - 5 min
- Expected Outcome be able to run an SVC algorithm with Intel Extension for Scikit

https://hprc.tamu.edu/events/workshops/
Partnering on Outreach

Leverage our programs to strengthen your broader impacts

- Teach a short course on computing to the TAMU community.
- Become an instructor in the Summer Computing Academy program camps for middle and high school students.
- Join the NSF SWEETER CyberTeam to explore computing-driven research and educational partnerships with universities in Texas, New Mexico, and Arizona.
- Participate in the NSF BRICCCs community to support research computing at smaller institutions and community colleges.
- Make your computing products available on the NSF ACES, NSF FASTER, and NSF Frontera machines.
- Mentor our students in international Student Cluster Competitions.