ACES Phase I
Next Generation Composability

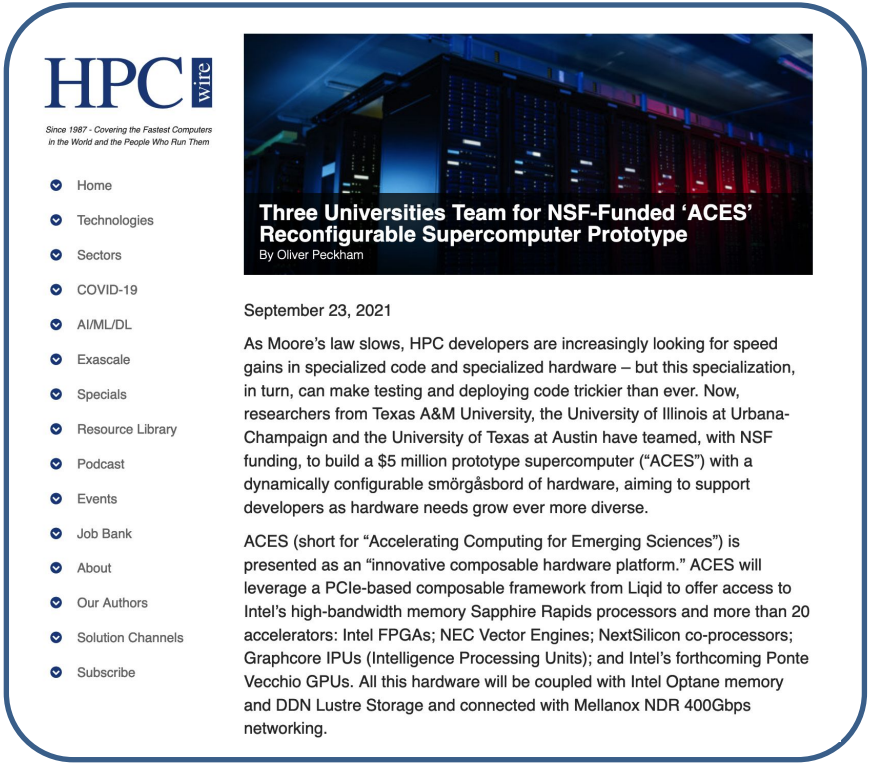
July 8, 2022

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

High Performance Research Computing
DIVISION OF RESEARCH

TACC
TEXAS ADVANCED COMPUTING CENTER

ACES - Accelerating Computing for Emerging Sciences



HPCwire

Since 1987 - Covering the Fastest Computers in the World and the People Who Run Them

- Home
- Technologies
- Sectors
- COVID-19
- AI/ML/DL
- Exascale
- Specials
- Resource Library
- Podcast
- Events
- Job Bank
- About
- Our Authors
- Solution Channels
- Subscribe

Three Universities Team for NSF-Funded 'ACES' Reconfigurable Supercomputer Prototype

By Oliver Peckham

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 smörgåsbord 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 composable hardware platform." ACES will leverage a PCIe-based composable framework from Liqid 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.

ACES
ACCELERATING COMPUTING FOR EMERGING SCIENCES

"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."

NSF

https://www.hpcwire.com/2021/09/23/three-universities-team-for-nsf-funded-aces-reconfigurable-supercomputer-prototype/

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.

ACES

ACCELERATING COMPUTING FOR EMERGING SCIENCES

ACES Project Team

Project Management Board:

- Honggao Liu (PI, Texas A&M University)
- Lisa Perez (Co-PI, Texas A&M University)
- Dhruva Chakravorty (Co-PI, Texas A&M University)
- Shaowen Wang (Co-PI, University of Illinois Urbana-Champaign)
- Timothy Cockerill (Co-PI, University of Texas Austin)
- Francis Dang (SI, Texas A&M University)
- Costas Georghiades (SI, Texas A&M University)
- Edwin Pierson (SI, Texas A&M University)

Advisory Council:

Gabrielle Allen (U. Wyoming), Richard Gerber (NERSC), John Goodhue (MGHPCC), Dan Katz(NCSA), Victor Hazlewood (U. Tennessee), Anita Nikolich (UIUC), Barry Schneider (NIST), Carol Song (Purdue), and Dan Stanzione (TACC).

NSF Program Officer: Robert Chadduck (NSF OAC Program Director)

A test-bed is Needed

“A large number of accelerators have become available to accelerate computing workloads. Researchers need access to them on a platform where they can test them!”

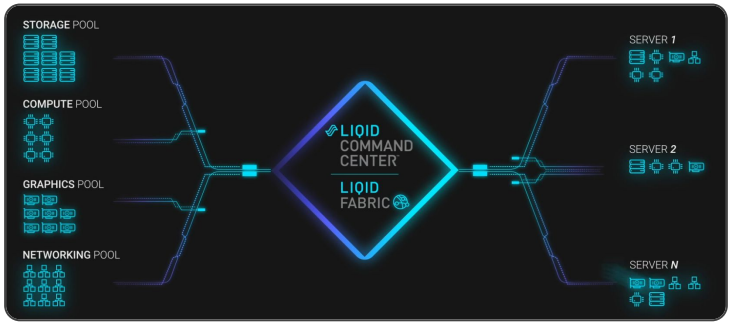
High Performance Computing (HPC) Architecture Comparison

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



A Modern HPC Platform Supporting Composable GPUs/Accelerators and Memory

Composability at the Hardware Level

FASTER*

Typical HPC layout

Composable layout

hprc.tamu.edu/resources

ACES - Accelerating Computing for Emerging Sciences (Phase I)

Component	Quantity	Description
Graphcore IPU	16	16 Colossus GC200 IPUs and dual AMD Rome CPU server on a 100 GbE RoCE fabric
Intel FPGA PAC D5005	2	FPGA SOC with Intel Stratix 10 SX FPGAs, 64 bit quad-core Arm Cortex-A53 processors, and 32GB DDR4
Intel Optane SSDs	8	3 TB of Intel Optane SSDs addressable as memory using MemVerge Memory Machine.

Available through FASTER (NSF Award #2019129)

ACES - Accelerating Computing for Emerging Sciences (Phase II)

Component	Quantity*	Description
Graphcore IPU	32	16 Colossus GC200 IPUs, 16 Bow IPUs, and a dual AMD Rome CPU server on a 100 GbE RoCE fabric
Intel FPGA PAC D5005	2	FPGA SOC with Intel Stratix 10 SX FPGAs, 64 bit quad core Arm Cortex-A53 processors, and 32GB DDR4
Bittware IA-840F FPGA	2	Accelerator based on Intel Agilex FPGA
NextSilicon coprocessor	20	Reconfigurable accelerator with an optimizer continuously evaluating application behavior.
NEC Vector Engine	24	Vector computing card (8 cores and HBM2 memory)
Intel Ponte Vecchio GPU	100	Intel GPUs for HPC, DL Training, AI Inference
Intel Optane SSDs	48	18 TB of Intel Optane SSDs addressable as memory w/ MemVerge Memory Machine.

*Estimated quantities

ACES System Description (Phase II)

Component	Quantity	Description
Allocatable resources		Total cores: 11,520
CPU-centric computing with variable memory requirements	120 nodes (11,520 cores)	Dual Intel Sapphire Rapids 2.1 GHz 48 core processors 96 cores per node, 512 GB memory, 1.6 TB NVMe storage (PCIe 5.0), NVIDIA Mellanox NDR 200 Gbps InfiniBand
Composable infrastructure	120 nodes	Dynamically reconfigurable infrastructure that allows up to 20 PCIe cards (GPU, FPGA, VE, etc.) per compute node
Data transfer nodes	2 nodes	Same as compute nodes, 100 Gbps network adapter

Research Workflows - Accelerators (Phases I and II)

Hardware Profile	Applications Supported	
NEC Vector Engines	• AI/ML (Statistical Machine Learning, Data Frame) • Chemistry (VASP, Quantum ESPRESSO) • Earth Sciences • NumPy Acceleration	• Oil & Gas (Seismic Imaging, Reservoir Simulation) • Plasma Simulation • Weather/Climate Simulation
Graphcore IPUs	• Graph Data • LSTM Neural Networks	• Markov Chain Monte Carlo • Natural Language Processing (Deep Learning)
Intel/Bittware FPGA	• AI Models for Embedded Use Cases • Big Data • CXL Memory Interface • Deep Learning Inference • Genomics	• MD Codes • Microcontroller Emulation for Autonomy Simulations • Streaming Data Analysis
Intel Optane SSDs	• Bioinformatics • Computational Fluid Dynamics (OpenFOAM)	• MD Codes • R • WRF
NextSilicon	• Biosciences (BLAST) • Computational Fluid Dynamics (OpenFOAM) • Cosmology (HACC) • Graph Search (Pathfinder)	• Molecular Dynamics (NAMD, AMBER, LAMMPS) • Quantum ChromoDynamics (MILC) • Weather/Environment modeling (WRF)

1,100+ Software Modules!

SOFTWARE MODULES ON THE FASTER CLUSTER

Last Updated: Jul 7 17:13:36 CDT

The available software for the Faster 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.

Show 10 entries Search: tensorflow

Name	Description
einops	'Flexible and powerful tensor operations for readable and reliable code. Supports numpy, pytorch, tensorflow, jax, and others.'
Horovod	'Horovod is a distributed training framework for TensorFlow.'
Keras	'Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. '
ONNX-Runtime	'ONNX Runtime inference can enable faster customer experiences and lower costs, supporting models from deep learning frameworks such as PyTorch and TensorFlow/Keras as well as classical machine learning libraries such as scikit-learn, LightGBM, XGBoost, etc. ONNX Runtime is compatible with different hardware, drivers, and operating systems, and provides optimal performance by leveraging hardware accelerators where applicable alongside graph optimizations and transforms.'

hprc.tamu.edu/software/faster/

Available Software Modules

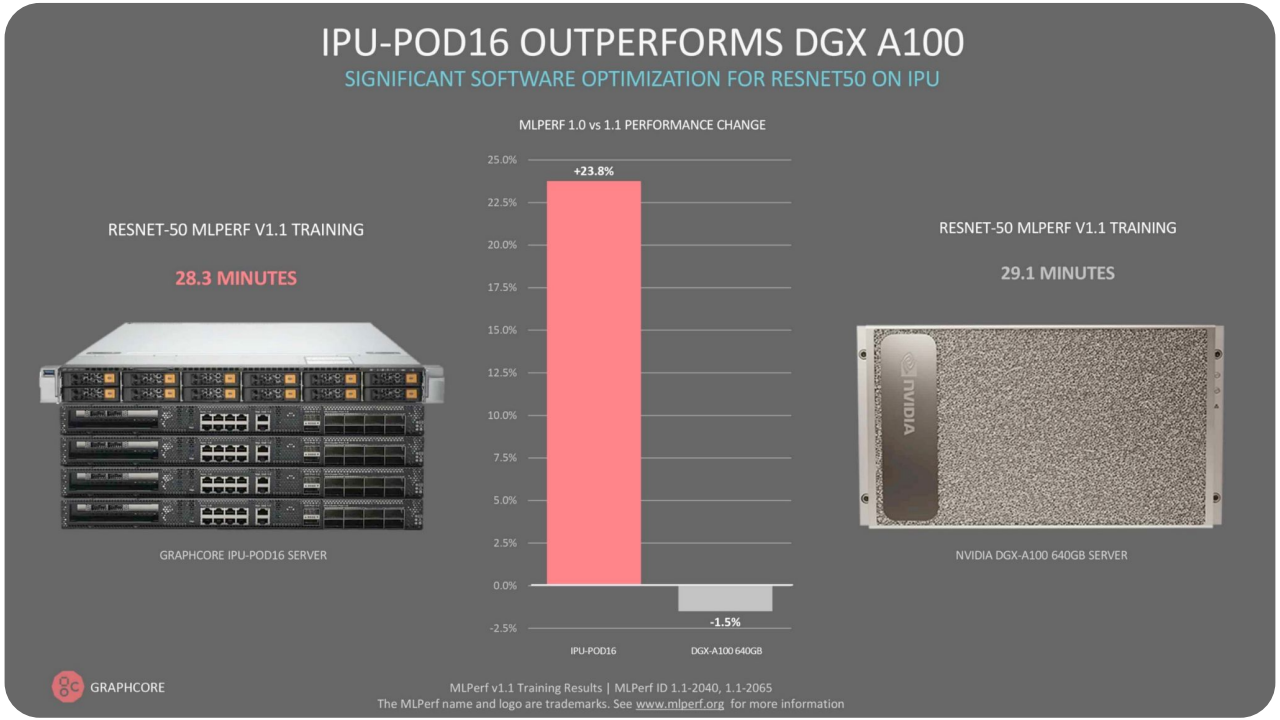
https://hprc.tamu.edu/wiki/SW:Modules

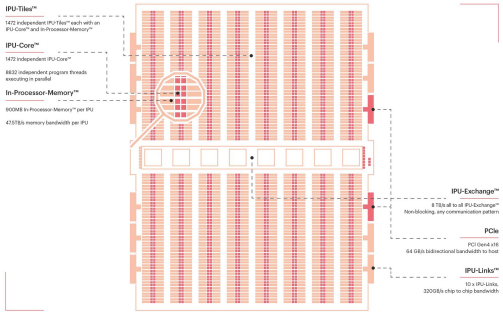
mla command to quickly search for installed software:

```
[username@faster1 ~]$ mla tensor
Using /home/username/module.avail.faster
Horovod/0.18.2-TensorFlow-1.15.2-Python-3.7.4
Horovod/0.22.1-CUDA-11.3.1-TensorFlow-2.6.0
TensorFlow/1.15.2-Python-3.7.4
TensorFlow/2.1.0-Python-3.7.4
TensorFlow/2.3.1-Python-3.8.2
TensorFlow/2.4.1
TensorFlow/2.5.0
TensorFlow/2.5.3-CUDA-11.3.1
TensorFlow/2.6.0-CUDA-11.3.1
TensorFlow/2.6.0
TensorFlow/2.7.1-CUDA-11.4.1
tensorboard/2.8.0
tensorboardX/2.2-PyTorch-1.7.1
tensorflow-probability/0.12.1
```

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

Graphcore IPUs (Intelligence Processing Unit)





https://www.graphcore.ai/posts/accelerating-resnet50-training-on-the-ipu-behind-our-mlperf-benchmark

GRAPHCORE

Porting TensorFlow 2 Models Quick Start

Version: Latest

Search docs

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

```
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

```
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

docs.graphcore.ai/en/latest/

Porting Tensorflow workflows to run on Graphcore IPUs

Viscous Burgers Equation: Physics-Informed Neural Networks for solving PDEs

An accurate, numerical approximation was used as a reference against which the PINN solution was measured. The PDE was constructed with sinusoidal initial condition and homogeneous Dirichlet boundary conditions as follows:

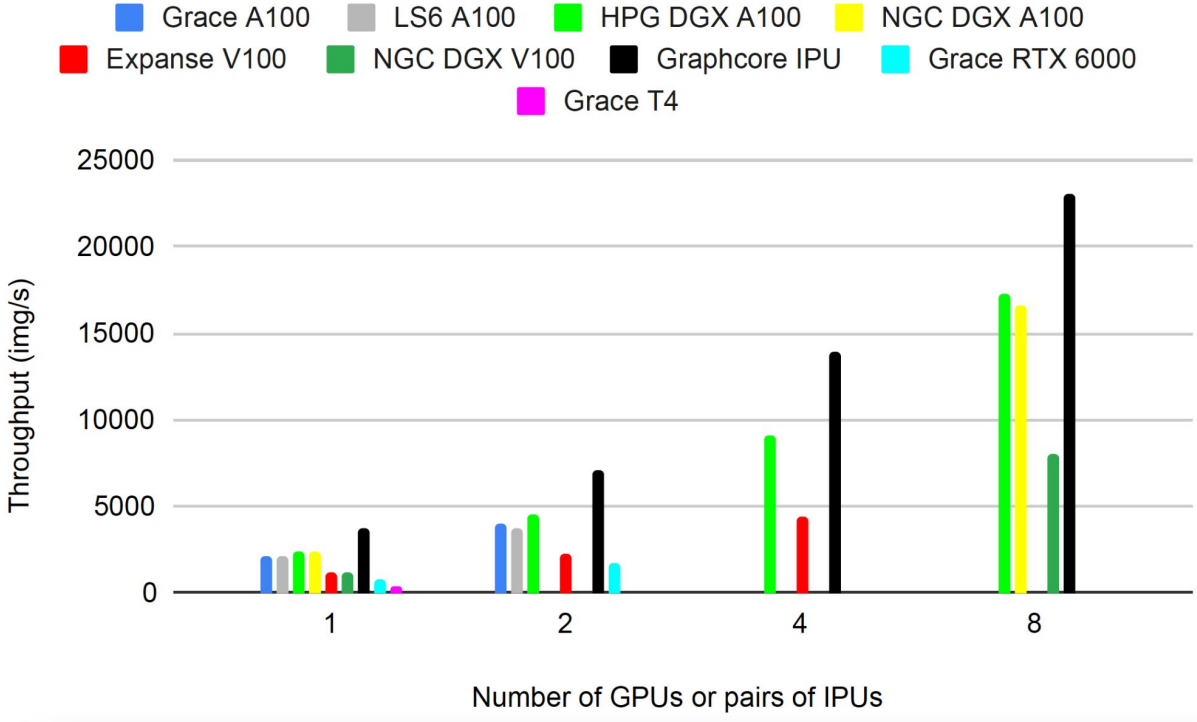
$$
\begin{aligned}
u_t + u \cdot u_x &= v \cdot u_{xx}, && (x,t) \in [-1,1] \times [0,\infty) \\
u(x,0) &= -\sin(\pi x), && -1 < x < 1, \\
u(1,t) &= u(-1,t) = 0, && t > 0.
\end{aligned}
$$

In the case where the fluid's viscosity, ***v***, is smaller than **~0.1π** a discontinuous shock-wave forms at **x=0**.

The PINN solution of the viscous Burgers' PDE was calculated using TensorDiffEq, an open-source TensorFlow 2.X-based package developed by researchers at Texas A&M University. This solution was found to be in excellent agreement with classic numerical solutions, with both solutions becoming unstable for very low viscosity, whilst taking less time to complete.

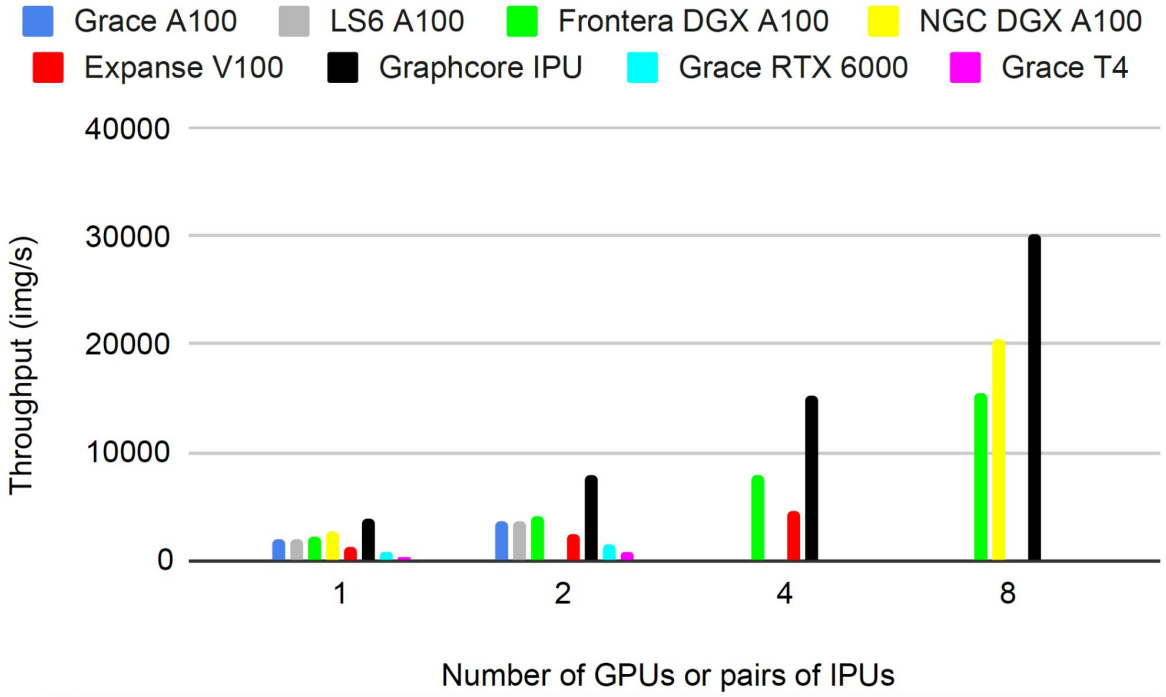
www.graphcore.ai/posts/ai-for-simulation-how-graphcore-is-helping-transform-traditional-hpc

PyTorch ResNet50 - GPU vs IPU



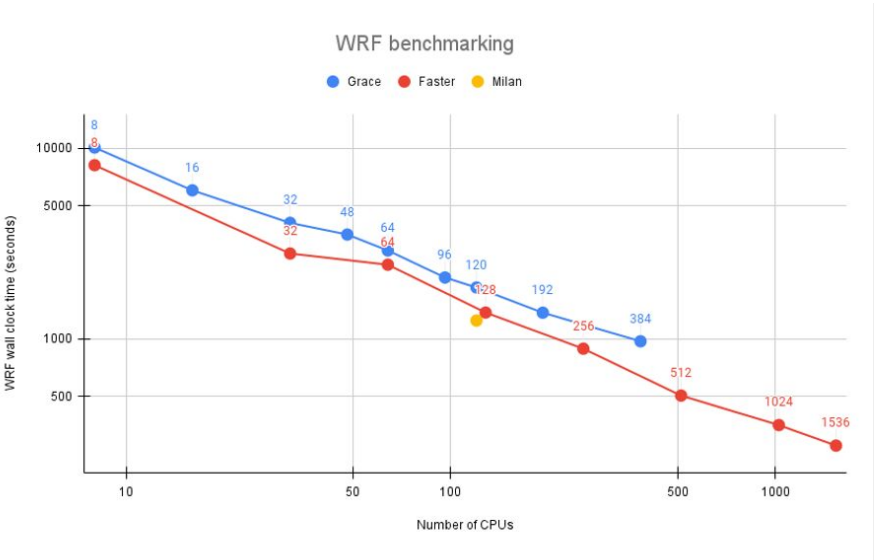
Abhinand S. Nasari, Hieu T. Le, Richard Lawrence, Zhenhua He, Xin Yang, Mario M. Krell, Alex Tsyplikhin, Mahidhar Tatineni, Tim Cockerill, Lisa M. Perez, Dhruva K. Chakravorty and Honggao Liu. 2022. Benchmarking the Performance of Accelerators on National Cyberinfrastructure Resources for Artificial Intelligence / Machine Learning Workloads. In Practice and Experience in Advanced Research Computing (PEARC '22), July 10-14, 2022, Boston, MA, USA. ACM, New York, NY, USA, 13 Pages. https://doi.org/10.1145/3491418.3530772

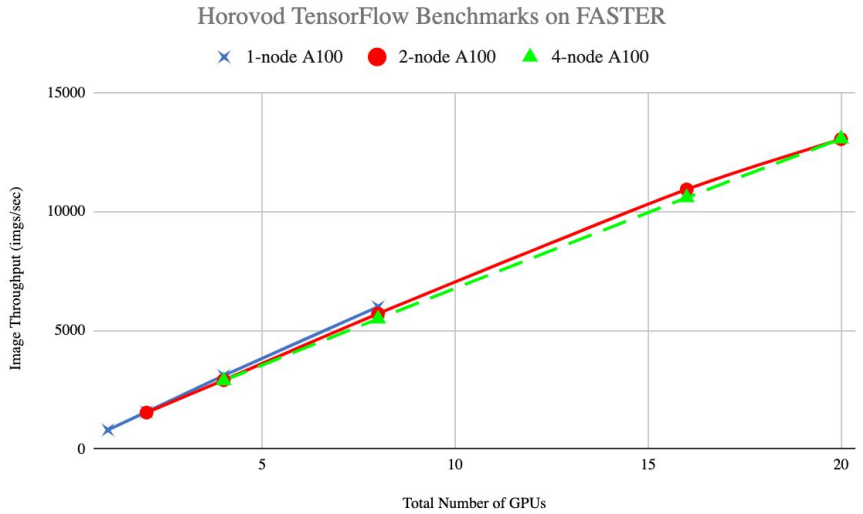
TF ResNet50 - GPU vs IPU



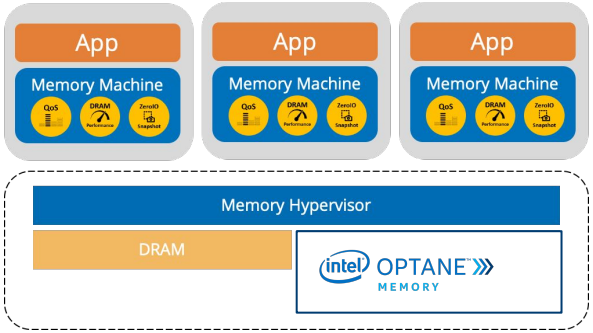
Abhinand S. Nasari, Hieu T. Le, Richard Lawrence, Zhenhua He, Xin Yang, Mario M. Krell, Alex Tsyplikhin, Mahidhar Tatineni, Tim Cockerill, Lisa M. Perez, Dhruva K. Chakravorty and Honggao Liu. 2022. Benchmarking the Performance of Accelerators on National Cyberinfrastructure Resources for Artificial Intelligence / Machine Learning Workloads. In Practice and Experience in Advanced Research Computing (PEARC '22), July 10-14, 2022, Boston, MA, USA. ACM, New York, NY, USA, 13 Pages. https://doi.org/10.1145/3491418.3530772

Scaling on Composable Fabrics





MemVerge Memory Machine



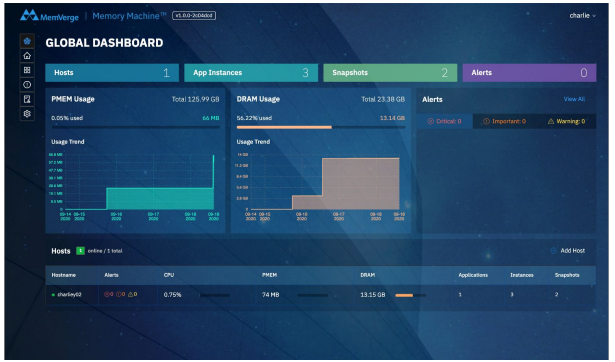
HPL Benchmark

HPL baseline without MemVerge Memory Machine:

T/V	N	NB	P	Q	Time	Gflops
WR12C2R4	140000	384	1	1	569.21	**3.21387e+03**

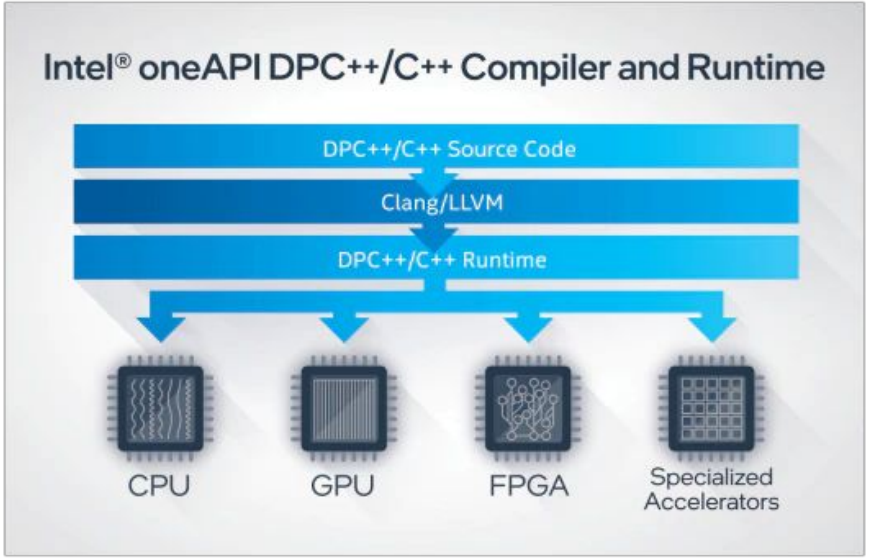
HPL with same input and memory usage above 140G going to SSDs with MemVerge Memory Machine:

T/V	N	NB	P	Q	Time	Gflops
WR12C2R4	140000	384	1	1	600.93	**3.04423e+03**



Intel FPGA PAC D5005

- Intel Stratix 10 SX FPGA family
  - Inline interfaces - 100 Gbps
- Fast code prototyping - OneAPI DPC++
  - Compile on CPU, validate design, then compile via FPGA
- Software
  - Intel Quartus Prime Pro
  - Intel FPGA SDK for OpenCL
  - Intel FPGA Add-On for the oneAPI Base Toolkit



Intel FPGA PAC D5005

FFTFPGA

FFTFPGA is an OpenCL based library for Fast Fourier Transformations for FPGAs. This repository provides OpenCL host code in the form of FFTW like APIs, which can be used to offload existing FFT routines to FPGAs with minimal effort. It also provides OpenCL kernels that can be synthesized to bitstreams, which the APIs can utilize.

Supported FPGAs

This library has been tested using the following FPGAs present in the Noctua cluster of the Paderborn Center for Parallel Computing (PC2) at Paderborn University:

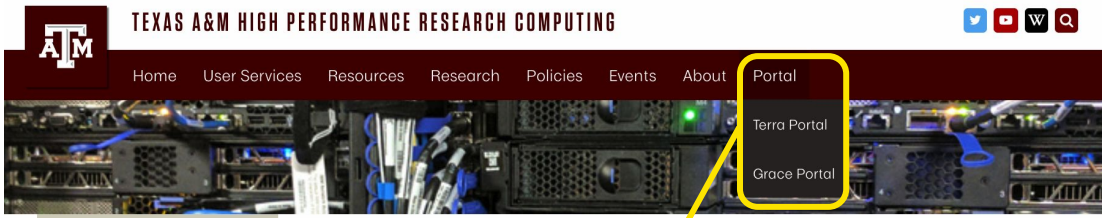
- Bittware 520N card with Intel Stratix 10 GX 2800 FPGA
- Intel FPGA PAC D5005 card with Intel Stratix 10 SX 2800 FPGA

Who is using FFTFPGA?

- CP2K: the quantum chemistry software package has an interface to offload 3d FFTs to Intel FPGAs that uses the OpenCL kernel designs of FFTFPGA.

github.com/pc2/fft3d-fpga

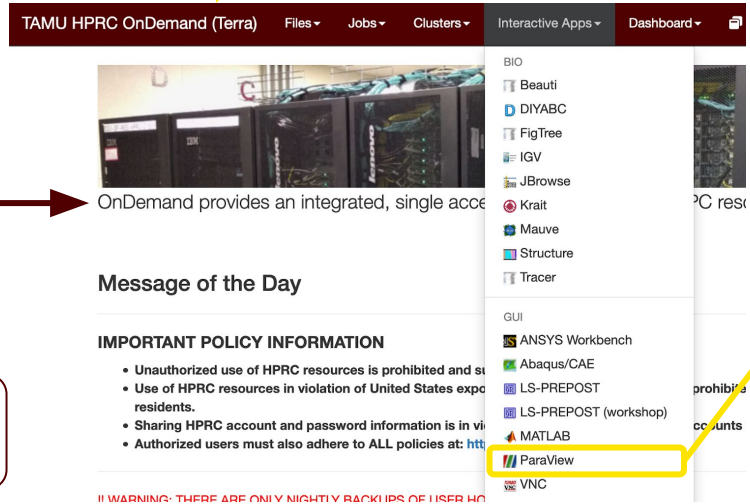
ACES Portal - OOD

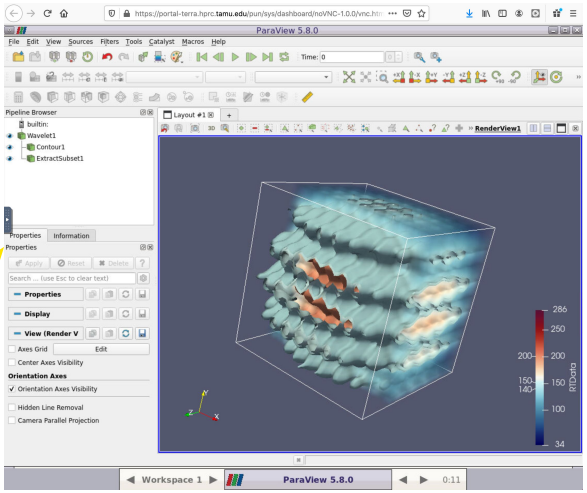


https://portal.hprc.tamu.edu

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

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





HPRC Portal YouTube tutorials

Knowledge Base

Graphcore IPUs [edit]

From one of FASTER login nodes, ssh into poplar1 system.

```
[username@faster1 ~]$ ssh poplar1
```

1.Set up the Poplar SDK environment [edit]

In this step, set up several environment variables to use the Graphcore tools and Poplar graph programming framework.

```
[username@poplar1 ~]$ source /opt/gc/poplar/poplar_sdk-ubuntu_18_04-[ver]/poplar-ubuntu_18_04-[ver]/enable.sh
[username@poplar1 ~]$ source /opt/gc/poplar/poplar_sdk-ubuntu_18_04-[ver]/popart-ubuntu_18_04-[ver]/enable.sh
```

[ver] indicates the version number of the package.

hprc.tamu.edu/wiki/ACES

Training Short Courses

Primers:

Linux
HPRC Clusters
Data Management
SLURM
Jupyter Notebook

Technology Lab:

Using AI Frameworks in Jupyter Notebook

Short Courses:

Python
Scientific Python
PyTorch
TensorFlow
MATLAB
Scientific ML
Julia
CUDA
Drug Docking
Quantum Chemistry
and more...

Short Courses:

NGS Analysis
NGS Metagenomics
NGS RADSeq/GBS
NGS Assembly
HPRC Galaxy
Linux
R
Perl
Fortran
OpenMP
MPI

Advanced Tech Training

Technology Lab: Using AI Frameworks in Jupyter Notebook

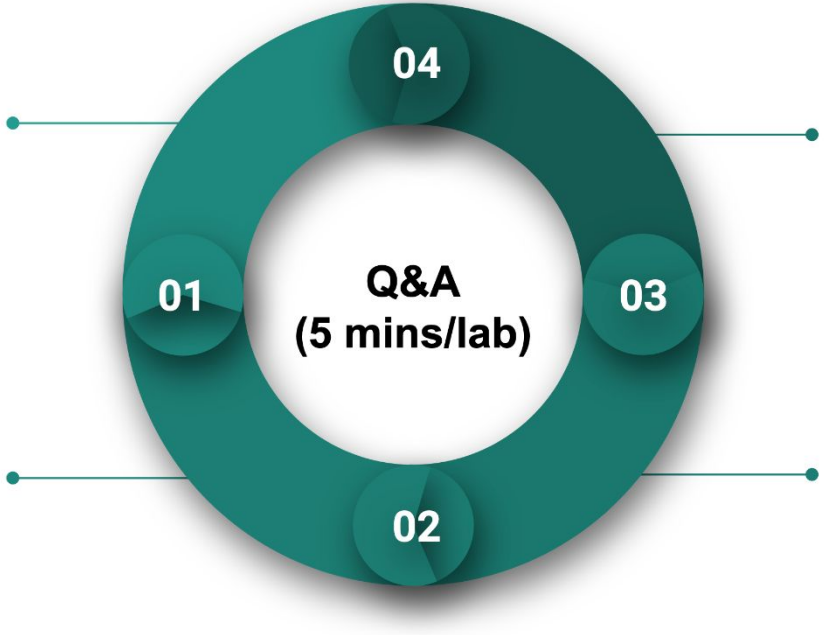
https://hprc.tamu.edu/training

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.



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).

Lab III Machine Learning (30 minutes)

We will learn to use scikit-learn library for linear regression and classification applications.

Industry Credentialed 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

Flyer

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 - 10 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 mi
  - Exercise complete with instructo – 20 mi
  - Exercise URL – https://github.com/mtolubaeva/numpy-tests
  - 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

https://hprc.tamu.edu/events/workshops/

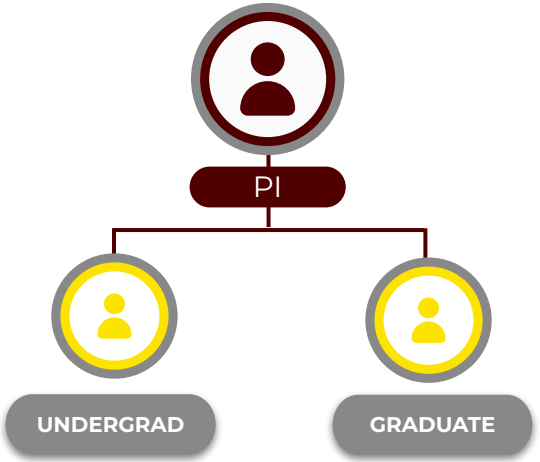
Onboarding Sessions

Join us on a zoom session:

Live support: M/W/F help sessions @ 8:30 AM CT

Getting on ACES Phase I

- Allocation is upon special request during this phase of deployment.
- You must have an XSEDE account!
- Applications are available at hprc.tamu.edu/aces/
- Email us at help@hprc.tamu.edu for questions, comments, and concerns



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

hprc.tamu.edu

HPRC Helpdesk:

help@hprc.tamu.edu

Phone: 979-845-0219