Exploiting Multi-/Many-core Intel CPUs for Deep Learning using MVAPICH2 MPI Library

Intel Booth Talk at ISC '19

by

Dhabaleswar K. (DK) Panda

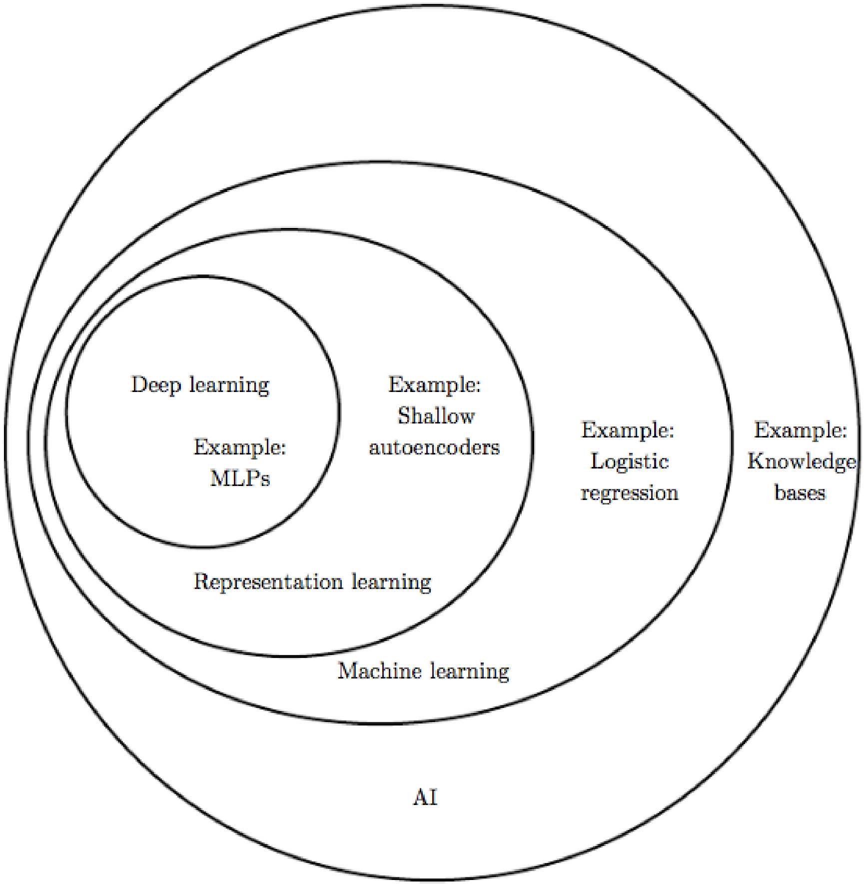
The Ohio State University

E-mail: panda@cse.ohio-state.edu

http://www.cse.ohio-state.edu/~panda

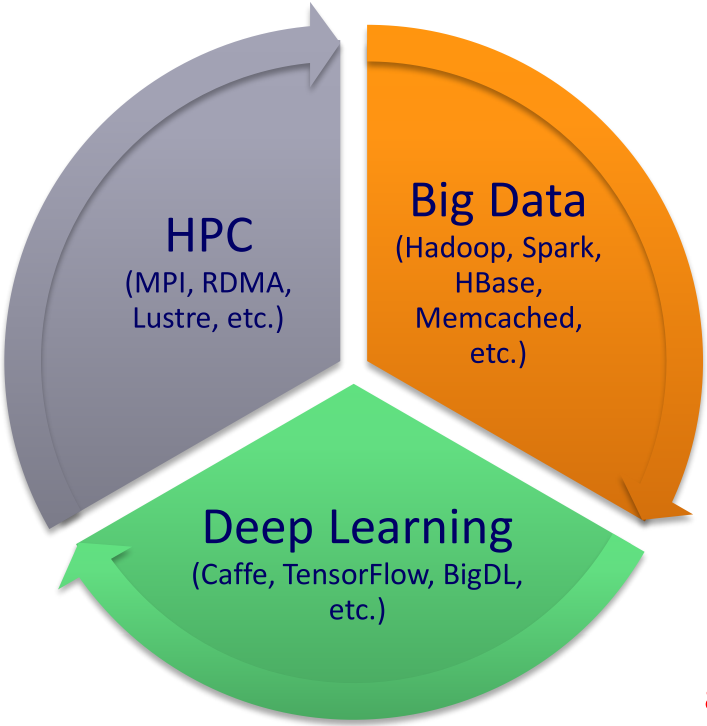
Understanding the Deep Learning Resurgence

- Deep Learning is a sub-set of Machine Learning
  – But, it is perhaps the most radical and revolutionary subset
  – Automatic feature extraction vs. hand-crafted features
- Deep Learning
  – A renewed interest and a lot of hype!
  – Key success: Deep Neural Networks (DNNs)
  – Everything was there since the late 80s except the "computability of DNNs"



Courtesy: http://www.deeplearningbook.org/contents/intro.html

Increasing Usage of HPC, Big Data and Deep Learning



Convergence of HPC, Big Data, and Deep Learning!

Increasing Need to Run these applications on the Cloud!!

Key Phases of Deep Learning

- Deep Learning has two major tasks
  1. Training of the Deep Neural Network
  2. Inference (or deployment) that uses a trained DNN
- DNN Training
  - Training is a compute/communication intensive process – can take days to weeks
  - Faster training is necessary!
- Faster training can be achieved by
  - Using Newer and Faster Hardware – But, there is a limit!
  - Can we use more GPUs or nodes?
    - The need for Parallel and Distributed Training

Broad Challenge: Exploiting HPC for Deep Learning

How to efficiently scale-out Deep Learning (DL) workloads by better exploiting High Performance Computing (HPC) resources like Multi-/Many-core CPUs?

Overview of the MVAPICH2 Project

- High Performance open-source MPI Library for InfiniBand, Omni-Path, Ethernet/iWARP, and RDMA over Converged Ethernet (RoCE)
  - MVAPICH (MPI-1), MVAPICH2 (MPI-2.2 and MPI-3.1), Started in 2001, First version available in 2002
  - MVAPICH2-X (MPI + PGAS), Available since 2011
  - Support for GPGPUs (MVAPICH2-GDR) and MIC (MVAPICH2-MIC), Available since 2014
  - Support for Virtualization (MVAPICH2-Virt), Available since 2015
  - Support for Energy-Awareness (MVAPICH2-EA), Available since 2015
  - Support for InfiniBand Network Analysis and Monitoring (OSU INAM) since 2015
  - **Used by more than 3,000 organizations in 88 countries**
  - **More than 545,000 (> 0.5 million) downloads from the OSU site directly**
  - Empowering many TOP500 clusters (Nov '18 ranking)
    - 3^{rd} ranked 10,649,640-core cluster (Sunway TaihuLight) at NSC, Wuxi, China
    - 14^{th}, 556,104 cores (Oakforest-PACS) in Japan
    - 17^{th}, 367,024 cores (Stampede2) at TACC
    - 27^{th}, 241,108-core (Pleiades) at NASA and many others
  - Available with software stacks of many vendors and Linux Distros (RedHat, SuSE, and OpenHPC)
  - **http://mvapich.cse.ohio-state.edu**
- Empowering Top500 systems for over a decade

Partner in the upcoming TACC Frontera System

Architecture of MVAPICH2 Software Family

High Performance Parallel Programming Models

- Message Passing Interface (MPI)
- PGAS (UPC, OpenSHMEM, CAF, UPC++)
- Hybrid --- MPI + X (MPI + PGAS + OpenMP/Cilk)

High Performance and Scalable Communication Runtime

Diverse APIs and Mechanisms

- Point-to-point Primitives
- Collectives Algorithms
- Job Startup
- Energy-Awareness
- Remote Memory Access
- I/O and File Systems
- Fault Tolerance
- Virtualization
- Active Messages
- Introspection & Analysis

Support for Modern Networking Technology (InfiniBand, iWARP, RoCE, Omni-Path)

Transport Protocols: RC, XRC, UD, DC

Modern Features: UMR, ODP, SR-IOV, Multi Rail

Support for Modern Multi-/Many-core Architectures (Intel-Xeon, OpenPower, Xeon-Phi, ARM, NVIDIA GPGPU)

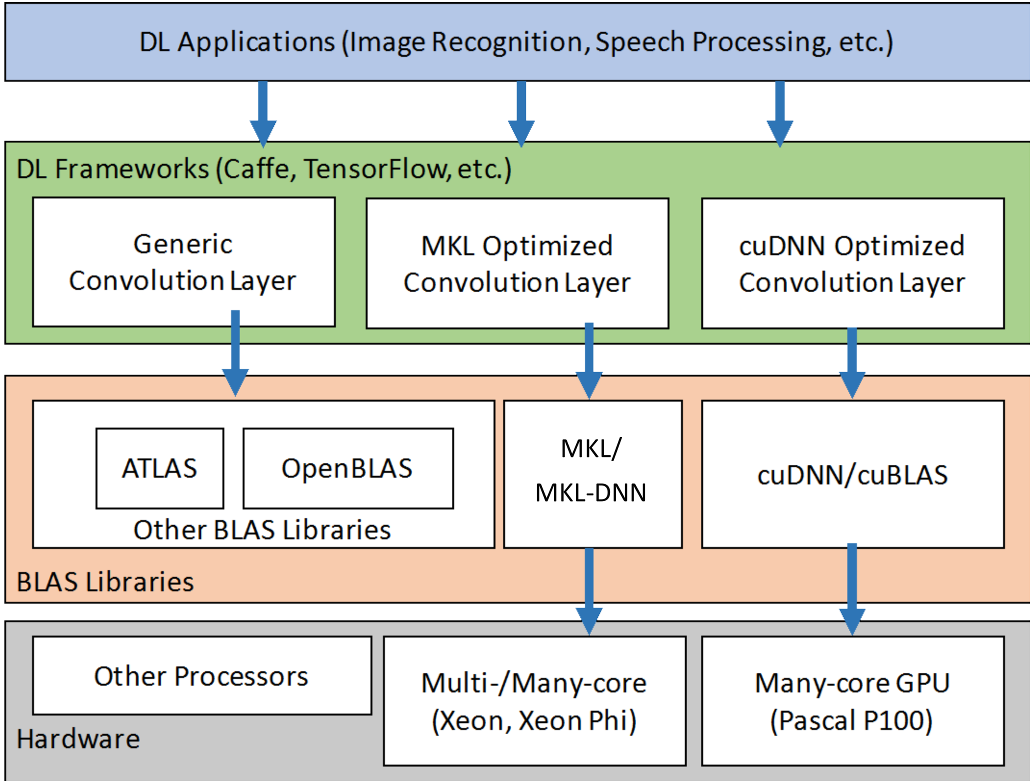
Transport Mechanisms: Shared Memory, CMA, IVSHMEM, XPMEM*

Modern Features: MCDRAM*, NVLink*, CAPI*

* Upcoming

Holistic Evaluation is Important!!

- My framework is faster than your framework!
- This needs to be understood in a holistic way.
- Performance depends on the entire execution environment (the full stack)
- Isolated view of performance is not helpful



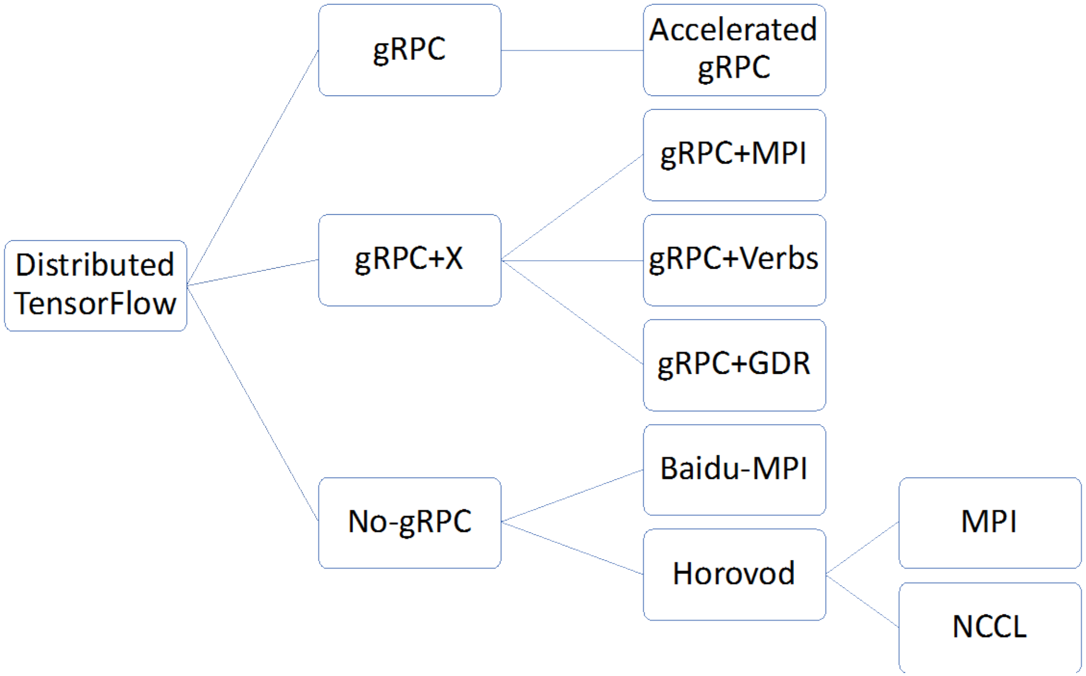
A. A. Awan, H. Subramoni, and Dhabaleswar K. Panda. "An In-depth Performance Characterization of CPU- and GPU-based DNN Training on Modern Architectures", In Proceedings of the Machine Learning on HPC Environments (MLHPC'17). ACM, New York, NY, USA, Article 8.

Three Key Insights

- Use Message Passing Interface (MPI) for single-node and multi-node training
  - Multi-process (MP) better than single-process (SP) approach
- Use Intel-optimized TensorFlow (MKL/MKL-DNN primitives)
  - Single-process (SP) training -- still under-optimized to fully utilize all CPU cores
- Overall performance depends on
  - Number of cores
  - Process per node (PPN) configuration
  - Hyper-threading (enabled/disabled)
  - DNN specifications like inherent parallelism between layers (inter-op parallelism)
  - Type of DNN (ResNet vs. Inception)

Distributed Training using TensorFlow (TF)

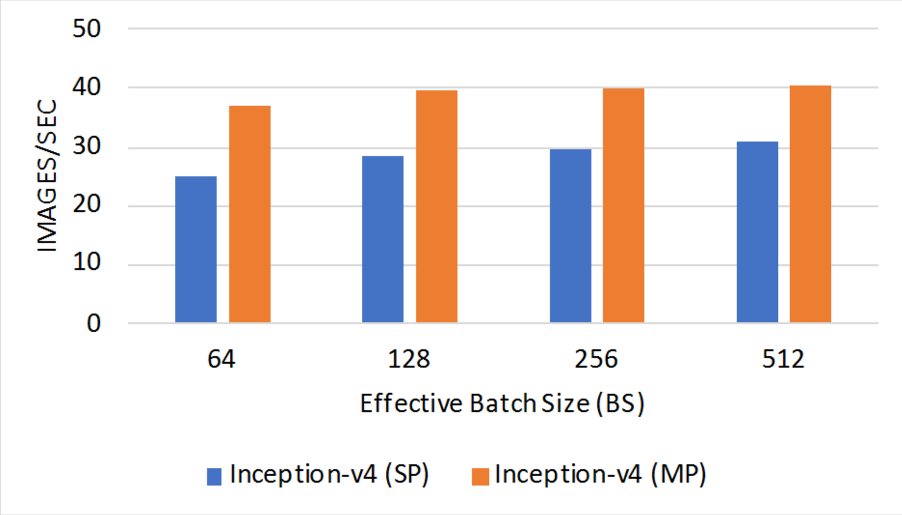
- TensorFlow is the most popular DL framework
- gRPC is the official distributed training runtime
  - Many problems for HPC use-cases
- Community efforts - Baidu and Uber's Horovod have added MPI support to TF across nodes
- Need to understand several options currently available →

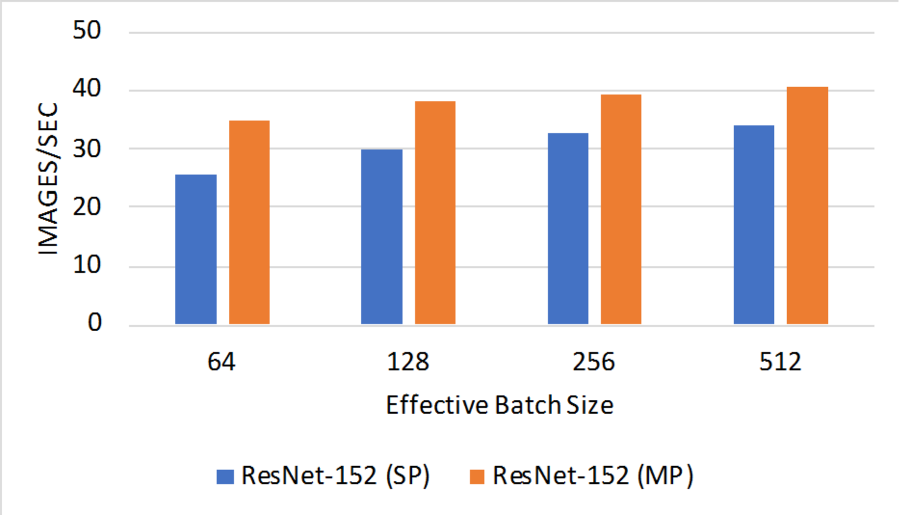


Awan et al., "Scalable Distributed DNN Training using TensorFlow and CUDA-Aware MPI: Characterization, Designs, and Performance Evaluation", CCGrid '19. https://arxiv.org/abs/1810.11112

Single-Process (SP) vs. Multi-Process (MP) on one node

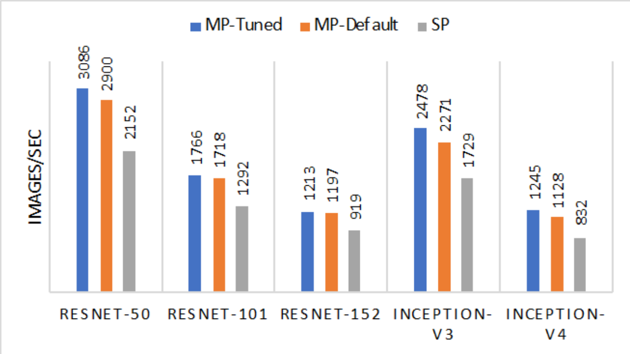
- Two different models on TACC Stampede (Intel Xeon Skylake – 48 cores)
- Key idea: MP is better than SP for all cases!
  - PPN and Hyper-threading needs to be tuned

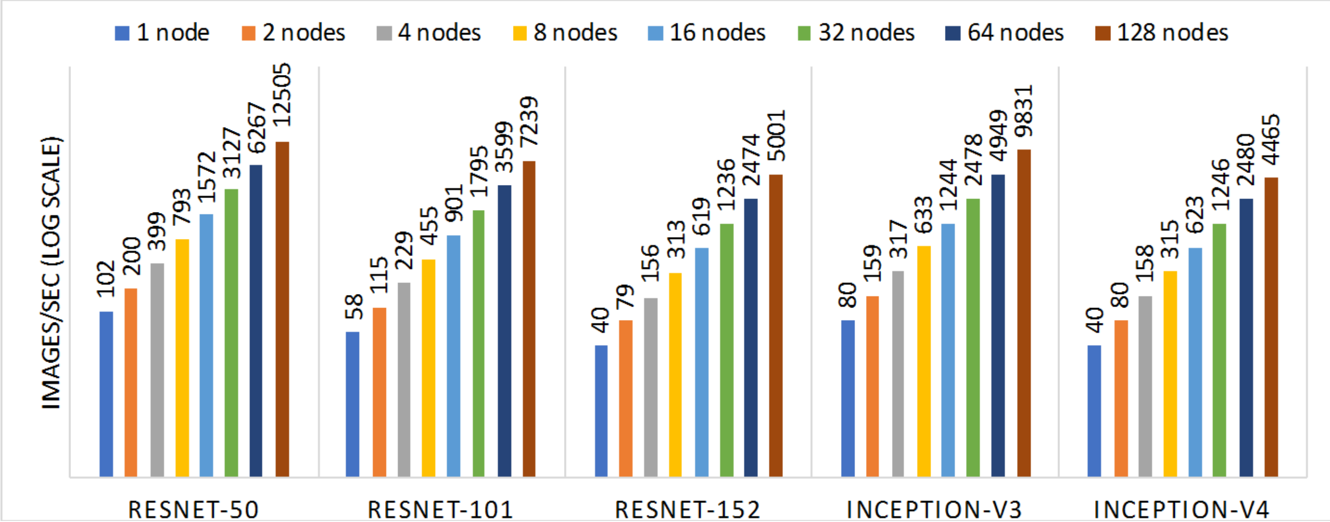




Multi-node Performance for TensorFlow

- Use tuned configuration (based on SP and MP) for multi-node →
  - PPN, batch size, and other parameters need to be tuned for best performance



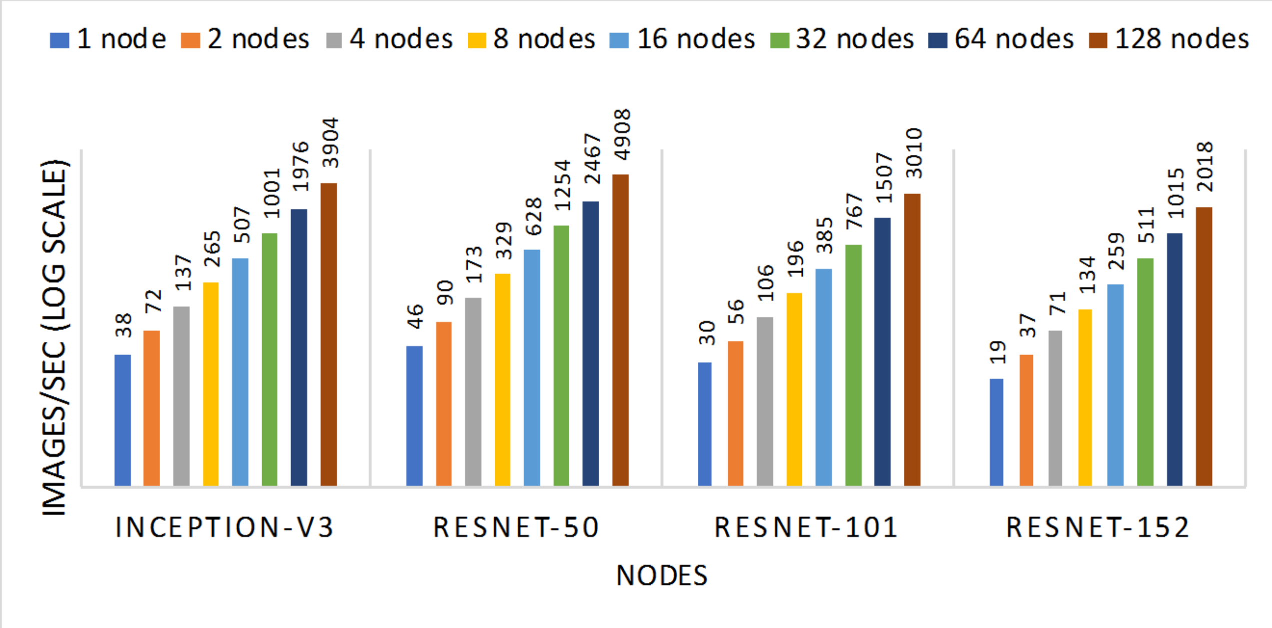


←

Using MVAPICH2, we achieved **125x speedup** (over single-node) on **128 nodes** for ResNet-152!

Multi-node Performance for PyTorch

- Early results with PyTorch (using tuned configuration)
  - Good scaling (106X speedup on 128 nodes)
  - Overall -- Slower than TensorFlow



Conclusions

- Scalable distributed training is getting important
- Requires high-performance middleware designs while exploiting modern interconnects
- Provided a set of different approaches to achieve scalable distributed training
  - Optimized collectives for CPU-based training
  - Using Intel-optimized DL frameworks
  - Use MPI for both single-node and multi-node training
- Will continue to enable the DL community to achieve scalability and high-performance for their distributed training workloads

Funding Acknowledgments

Funding Support by

Office of Science U.S. Department of Energy
NSF
Mellanox Technologies
Microsoft
QLogic
arm
Cray The Supercomputer Company
intel
Linux Networx
NVIDIA
NetApp
Sun microsystems
Cisco Systems

Equipment Support by

Mellanox Technologies
intel
arm
AMD
Sun microsystems
Appro
advanced clustering technologies, inc.
Microway Technology you can count on
NVIDIA
QLogic

Personnel Acknowledgments

Current Students (Graduate)

- A. Awan (Ph.D.)
- M. Bayatpour (Ph.D.)
- S. Chakraborthy (Ph.D.)
- C.-H. Chu (Ph.D.)
- J. Hashmi (Ph.D.)
- A. Jain (Ph.D.)
- K. S. Kandadi (M.S.)
- K. S. Khorassani (Ph.D.)
- P. Kousha (Ph.D.)
- A. Quentin (Ph.D.)
- B. Ramesh (M. S.)
- D. Shankar (Ph.D.)
- S. Xu (M.S.)
- Q. Zhou (Ph.D.)

Current Students (Undergraduate)

- V. Gangal (B.S.)
- N. Sarkauskas (B.S.)
- A. Yeretzian (B.S.)

Current Research Scientist

- H. Subramoni

Current Post-docs

- M. S. Ghazimeersaeed
- A. Ruhela
- K. Manian

Current Research Specialist

- J. Smith

Past Students

- A. Augustine (M.S.)
- P. Balaji (Ph.D.)
- R. Biswas (M.S.)
- S. Bhagvat (M.S.)
- A. Bhat (M.S.)
- D. Buntinas (Ph.D.)
- L. Chai (Ph.D.)
- B. Chandrasekharan (M.S.)
- N. Dandapanthula (M.S.)
- V. Dhanraj (M.S.)
- T. Gangadharappa (M.S.)
- K. Gopalakrishnan (M.S.)
- W. Huang (Ph.D.)
- W. Jiang (M.S.)
- J. Jose (Ph.D.)
- S. Kini (M.S.)
- M. Koop (Ph.D.)
- K. Kulkarni (M.S.)
- R. Kumar (M.S.)
- S. Krishnamoorthy (M.S.)
- K. Kandalla (Ph.D.)
- M. Li (Ph.D.)
- P. Lai (M.S.)
- J. Liu (Ph.D.)
- M. Luo (Ph.D.)
- A. Mamidala (Ph.D.)
- G. Marsh (M.S.)
- V. Meshram (M.S.)
- A. Moody (M.S.)
- S. Naravula (Ph.D.)
- R. Noronha (Ph.D.)
- X. Ouyang (Ph.D.)
- S. Pai (M.S.)
- S. Potluri (Ph.D.)
- R. Rajachandrasekar (Ph.D.)
- G. Santhanaraman (Ph.D.)
- A. Singh (Ph.D.)
- J. Sridhar (M.S.)
- S. Sur (Ph.D.)
- H. Subramoni (Ph.D.)
- K. Vaidyanathan (Ph.D.)
- A. Vishnu (Ph.D.)
- J. Wu (Ph.D.)
- W. Yu (Ph.D.)
- J. Zhang (Ph.D.)

Past Research Scientist

- K. Hamidouche
- S. Sur
- X. Lu

Past Programmers

- D. Bureddy
- J. Perkins

Past Research Specialist

- M. Arnold

Past Post-Docs

- D. Banerjee
- X. Besseron
- H.-W. Jin
- J. Lin
- M. Luo
- E. Mancini
- S. Marcarelli
- J. Vienne
- H. Wang

Thank You!

panda@cse.ohio-state.edu

Network Based Computing Laboratory

Network-Based Computing Laboratory
http://nowlab.cse.ohio-state.edu/

MVAPICH
MPI, PGAS and Hybrid MPI+PGAS Library

The High-Performance MPI/PGAS Project
http://mvapich.cse.ohio-state.edu/

HiBD
High-Performance Big Data

The High-Performance Big Data Project
http://hibd.cse.ohio-state.edu/

HiDL
High-Performance Deep Learning

The High-Performance Deep Learning Project
http://hidl.cse.ohio-state.edu/