Middleware for Message Passing Interface (MPI) and Deep Learning on OpenPOWER platforms

OpenPOWER ADG Workshop (Nov ’20)

by

Dhabaleswar K. (DK) Panda

The Ohio State University

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

http://www.cse.ohio-state.edu/~panda
High-End Computing (HEC): PetaFlop to ExaFlop

100 PetaFlops in 2017

415 PetaFlops in 2020
(Fugaku in Japan with 7.3M cores)

1 ExaFlops

Expected to have an ExaFlop system in 2021!
Increasing Usage of HPC, Big Data and Deep/Machine Learning

Convergence of HPC, Big Data, and Deep/Machine Learning!

Increasing Need to Run these applications on the Cloud!!
Presentation Overview

• MVAPICH Project
  – MPI and PGAS (MVAPICH) Library with CUDA-Awareness

• HiDL Project
  – High-Performance Deep Learning

• Conclusions
Overview of the MVAPICH2 Project

- High Performance open-source MPI Library
- Support for multiple interconnects
  - InfiniBand, Omni-Path, Ethernet/iWARP, RDMA over Converged Ethernet (RoCE), and AWS EFA
- Support for multiple platforms
  - x86, OpenPOWER, ARM, Xeon-Phi, GPGPUs (NVIDIA and AMD (upcoming))
- Started in 2001, first open-source version demonstrated at SC ‘02
- Supports the latest MPI-3.1 standard
- http://mvapich.cse.ohio-state.edu
- Additional optimized versions for different systems/environments:
  - MVAPICH2-X (Advanced MPI + PGAS), since 2011
  - MVAPICH2-GDR with support for NVIDIA GPGPUs, since 2014
  - MVAPICH2-MIC with support for Intel Xeon-Phi, since 2014
  - MVAPICH2-Virt with virtualization support, since 2015
  - MVAPICH2-EA with support for Energy-Awareness, since 2015
  - MVAPICH2-Azure for Azure HPC IB instances, since 2019
  - MVAPICH2-X-AWS for AWS HPC+EFA instances, since 2019
- Tools:
  - OSU MPI Micro-Benchmarks (OMB), since 2003
  - OSU InfiniBand Network Analysis and Monitoring (INAM), since 2015
- Used by more than 3,100 organizations in 89 countries
- More than 1 million downloads from the OSU site directly
- Empowering many TOP500 clusters (June ‘20 ranking)
  - 4th, 10,649,600-core (Sunway TaihuLight) at NSC, Wuxi, China
  - 8th, 448, 448 cores (Frontera) at TACC
  - 12th, 391,680 cores (ABCI) in Japan
  - 18th, 570,020 cores (Nurion) in South Korea and many others
- Available with software stacks of many vendors and Linux Distro (RedHat, SuSE, OpenHPC, and Spack)
- Partner in the 8th ranked TACC Frontera system
- Empowering Top500 systems for more than 15 years
Architecture of MVAPICH2 Software Family for HPC and DL/ML

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, Elastic Fabric Adapter)

- Transport Protocols: RC, SRD, 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: Optane*, NVLink, CAPI*

* Upcoming
## MVAPICH2 Software Family

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Library</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPI with IB, iWARP, Omni-Path, and RoCE</td>
<td>MVAPICH2</td>
</tr>
<tr>
<td>Advanced MPI Features/Support, OSU INAM, PGAS and MPI+PGAS with IB, Omni-Path, and RoCE</td>
<td>MVAPICH2-X</td>
</tr>
<tr>
<td>MPI with IB, RoCE &amp; GPU and Support for Deep/Machine Learning</td>
<td>MVAPICH2-GDR</td>
</tr>
<tr>
<td>HPC Cloud with MPI &amp; IB</td>
<td>MVAPICH2-Virt</td>
</tr>
<tr>
<td>Energy-aware MPI with IB, iWARP and RoCE</td>
<td>MVAPICH2-EA</td>
</tr>
<tr>
<td>MPI Energy Monitoring Tool</td>
<td>OEMT</td>
</tr>
<tr>
<td>InfiniBand Network Analysis and Monitoring</td>
<td>OSU INAM</td>
</tr>
<tr>
<td>Microbenchmarks for Measuring MPI and PGAS Performance</td>
<td>OMB</td>
</tr>
</tbody>
</table>
Converged Middleware for HPC, Big Data and Deep/Machine Learning

- **HPC**
  - (MPI, PGAS, etc.)

- **Big Data**
  - (Hadoop, Spark, HBase, Memcached, etc.)

- **Deep/Machine Learning**
  - (TensorFlow, PyTorch, BigDL, cuML, etc.)

Network Based Computing Laboratory

OpenPOWER-ADG (Nov ‘20)
Intra-node Point-to-Point Performance on OpenPOWER

**Intra-Socket Small Message Latency**

- **MVAPICH2 2.3.5**
- **SpectrumMPI-10.3.1.00**

**Intra-Socket Large Message Latency**

- **MVAPICH2 2.3.5**
- **SpectrumMPI-10.3.1.00**

**Intra-Socket Bandwidth**

- **MVAPICH2 2.3.5**
- **SpectrumMPI-10.3.1.00**

**Intra-Socket Bi-directional Bandwidth**

- **MVAPICH2 2.3.5**
- **SpectrumMPI-10.3.1.00**

Platform: Two nodes of OpenPOWER (Power9-ppc64le) CPU using Mellanox EDR (MT4121) HCA

MVAPICH2 2.3.5 (upcoming)
### Inter-node Point-to-Point Performance on OpenPower

#### Small Message Latency

<table>
<thead>
<tr>
<th></th>
<th>MVAPICH2 2.3.5</th>
<th>SpectrumMPI-10.3.1.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latency</td>
<td><img src="image1.png" alt="Graph" /></td>
<td></td>
</tr>
</tbody>
</table>

#### Large Message Latency

<table>
<thead>
<tr>
<th></th>
<th>MVAPICH2 2.3.5</th>
<th>SpectrumMPI-10.3.1.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latency</td>
<td><img src="image2.png" alt="Graph" /></td>
<td></td>
</tr>
</tbody>
</table>

#### Bandwidth

<table>
<thead>
<tr>
<th></th>
<th>MVAPICH2 2.3.5</th>
<th>SpectrumMPI-10.3.1.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bandwidth</td>
<td><img src="image3.png" alt="Graph" /></td>
<td></td>
</tr>
</tbody>
</table>

#### Bi-directional Bandwidth

<table>
<thead>
<tr>
<th></th>
<th>MVAPICH2 2.3.5</th>
<th>SpectrumMPI-10.3.1.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bi-Bandwidth</td>
<td><img src="image4.png" alt="Graph" /></td>
<td></td>
</tr>
</tbody>
</table>

*Platform: Two nodes of OpenPOWER (POWER9-ppc64le) CPU using Mellanox EDR (MT4121) HCA*
Optimized MVAPICH2 All-Reduce & Reduce on OpenPower

**All_Reduce**

![Graph showing latency for All_Reduce operations.](image)

**Reduce**

![Graph showing latency for Reduce operations.](image)
GPU-Aware (CUDA-Aware) MPI Library: MVAPICH2-GPU

- Standard MPI interfaces used for unified data movement
- Takes advantage of Unified Virtual Addressing (>= CUDA 4.0)
- Overlaps data movement from GPU with RDMA transfers

At Sender:

\[
\text{MPI\_Send(s\_devbuf, size, \ldots);}
\]

At Receiver:

\[
\text{MPI\_Recv(r\_devbuf, size, \ldots);}
\]

High Performance and High Productivity
Optimized MVAPICH2-GDR Design

**GPU-GPU Inter-node Latency**

- **Message Size (Bytes)**: 0, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1K, 2K, 4K
- **Latency (us)**: 0, 10, 20, 30
- **Bandwidth (MB/s)**: 0, 2000, 4000, 6000

- **MV2-(NO-GDR)**
- **MV2-GDR-2.3**

**GPU-GPU Inter-node Bandwidth**

- **Message Size (Bytes)**: 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1K, 2K, 4K
- **Bandwidth (MB/s)**: 0, 1000, 2000, 3000, 4000, 5000, 6000

- **MV2-(NO-GDR)**
- **MV2-GDR-2.3**

- **Latency (us)**: 1.85us, 10x
- **Bandwidth (MB/s)**: 9x

Intel Haswell (E5-2687W @ 3.10 GHz) node - 20 cores
NVIDIA Volta V100 GPU
Mellanox Connect-X4 EDR HCA
CUDA 9.0
Mellanox OFED 4.0 with GPU-Direct-RDMA
D-to-D Performance on OpenPOWER w/ GDRCopy (NVLink2 + Volta)

**Intra-Node Latency (Small Messages)**

*Latency (us) vs Message Size (Bytes)*

*Intra-node Latency: 0.76 us (with GDRCopy)*

**Intra-Node Latency (Large Messages)**

*Latency (us) vs Message Size (Bytes)*

**Intra-Node Bandwidth**

*Bandwidth (MB/s) vs Message Size (Bytes)*

*Intra-node Bandwidth: 65.48 GB/sec for 4MB (via NVLINK2)*

**Inter-Node Latency (Small Messages)**

*Latency (us) vs Message Size (Bytes)*

*Inter-node Latency: 2.18 us (with GDRCopy 2.0)*

**Inter-Node Latency (Large Messages)**

*Latency (us) vs Message Size (Bytes)*

**Inter-Node Bandwidth**

*Bandwidth (MB/s) vs Message Size (Bytes)*

*Inter-node Bandwidth: 23 GB/sec for 4MB (via 2 Port EDR)*

Platform: OpenPOWER (POWER9-ppc64le) nodes equipped with a dual-socket CPU, 4 Volta V100 GPUs, and 2port EDR InfiniBand Interconnect
Impact on HOOMD-blue, a Molecular Dynamics Application

**64K Particles**

- Number of Processes: 4, 8, 16, 32
- Average Time Steps per second (TPS): 0, 4, 8, 12
- Graph shows a 2x increase in performance with MV2+GDR compared to MV2.

**256K Particles**

- Number of Processes: 4, 8, 16, 32
- Average Time Steps per second (TPS): 0, 4, 8, 12
- Graph shows a 2x increase in performance with MV2+GDR compared to MV2.

Network Based Computing Laboratory

OpenPOWER-ADG (Nov ‘20)
Application-Level Evaluation (Cosmo) and Weather Forecasting in Switzerland

Wilkes GPU Cluster

CSCS GPU cluster

- 2X improvement on 32 GPUs nodes
- 30% improvement on 96 GPU nodes (8 GPUs/node)

On-going collaboration with CSCS and MeteoSwiss (Switzerland) in co-designing MV2-GDR and Cosmo Application


Cosmo model: http://www2.cosmo-model.org/content/tasks/operational/meteoSuisse/
Presentation Overview

• MVAPICH Project
  – MPI and PGAS (MVAPICH) Library with CUDA-Awareness

• HiDL Project
  – High-Performance Deep Learning

• Conclusions
**Scale-up and Scale-out**

- **Scale-up**: Intra-node Communication
  - Many improvements like:
    - NVIDIA cuDNN, cuBLAS, NCCL, etc.
    - CUDA Co-operative Groups

- **Scale-out**: Inter-node Communication
  - DL Frameworks – most are optimized for single-node only
  - Distributed (Parallel) Training is an emerging trend
    - PyTorch – MPI/NCCL2
    - TensorFlow – gRPC-based/MPI/NCCL2
    - OSU-Caffe – MPI-based
Broad Challenge: Exploiting HPC for DL and ML

How to efficiently Scale-up and scale-out Deep Learning (DL) frameworks and take advantage of heterogeneous High Performance Computing (HPC) resources?
MVAPICH2 (MPI)-driven Infrastructure for ML/DL Training

ML/DL Applications

- TensorFlow
- PyTorch
- MXNet

Horovod

MVAPICH2 or MVAPICH2-X for CPU Training

MVAPICH2-GDR for GPU Training

ML/DL Applications

- PyTorch

Torch.distributed

DeepSpeed

MVAPICH2 or MVAPICH2-X for CPU Training

MVAPICH2-GDR for GPU Training

More details available from: http://hidl.cse.ohio-state.edu
Accelerating DL and ML Applications with MVAPICH2 MPI Libraries

- MPI-driven Deep Learning
  - CPU-based Deep Learning
  - GPU-based Deep Learning
- Out-of-core DNN training
- Exploiting Hybrid (Data and Model) Parallelism
- Use-Case: AI-Driven Digital Pathology
- Commercial Support and Products
Distributed TensorFlow on TACC Frontera (2,048 CPU nodes)

• Scaled TensorFlow to 2048 nodes on Frontera using MVAPICH2 and IntelMPI

• MVAPICH2 and IntelMPI give similar performance for DNN training

• Report a peak of 260,000 images/sec on 2,048 nodes

• On 2048 nodes, ResNet-50 can be trained in 7 minutes!

Optimized designs in MVAPICH2-GDR offer better performance for most cases

MPI_Allreduce (MVAPICH2-GDR) vs. ncclAllreduce (NCCL2) up to 1,536 GPUs

Platform: Dual-socket IBM POWER9 CPU, 6 NVIDIA Volta V100 GPUs, and 2-port InfiniBand EDR Interconnect

Distributed TensorFlow on ORNL Summit (1,536 GPUs)

- ResNet-50 Training using TensorFlow benchmark on SUMMIT -- 1536 Volta GPUs!
- 1,281,167 (1.2 mil.) images
- Time/epoch = 3 seconds
- Total Time (90 epochs) = 3 x 90 = 270 seconds = 4.5 minutes!

*We observed issues for NCCL2 beyond 384 GPUs

Platform: The Summit Supercomputer (#2 on Top500.org) – 6 NVIDIA Volta GPUs per node connected with NVLink, CUDA 10.1

ImageNet-1k has 1.2 million images
MVAPICH2-GDR reaching ~0.42 million images per second for ImageNet-1k!
Scaling PyTorch on ORNL Summit using MVAPICH2-GDR

- ResNet-50 training using PyTorch + Horovod on Summit
  - Synthetic ImageNet dataset
  - Up to 256 nodes, 1536 GPUs

- MVAPICH2-GDR can outperform NCCL2
  - Up to 30% higher throughput

- CUDA 10.1 cuDNN 7.6.5
  - PyTorch v1.5.0 Horovod v0.19.1


Platform: The Summit Supercomputer (#2 on Top500.org) – 6 NVIDIA Volta GPUs per node connected with NVLink, CUDA 10.1
Scaling PyTorch with Horovod

- ResNet-50 training using PyTorch + Horovod on Lassen
  - Synthetic ImageNet dataset
  - Up to 64 nodes, 256 GPUs
- MVAPICH2-GDR can outperform NCCL2
  - Up to 15% higher throughput
- CUDA 10.2 cuDNN 7.6.5
  PyTorch v1.5.0 Horovod v0.19.4

Platform: The Lassen Supercomputer (#14 on Top500.org) – 4 NVIDIA Volta GPUs per node connected with NVLink, CUDA 10.2
Scaling PyTorch with DeepSpeed

- ResNet-50 training using PyTorch + DeepSpeed on Lassen
  - Synthetic ImageNet dataset
  - Up to 64 nodes, 256 GPUs

- MVAPICH2-GDR can outperform NCCL2
  - Up to 10% higher throughput

- CUDA 10.2 cuDNN 7.6.5
  PyTorch v1.5.0 DeepSpeed v0.2.0

Platform: The Lassen Supercomputer (#14 on Top500.org) – 4 NVIDIA Volta GPUs per node connected with NVLink, CUDA 10.2
Scaling PyTorch with `torch.distributed`

- ResNet-50 training using PyTorch + `torch.distributed` on Lassen
  - Synthetic ImageNet dataset
  - Up to 64 nodes, 256 GPUs
- MVAPICH2-GDR can outperform NCCL2
  - Up to 16% higher throughput
- CUDA 10.2 cuDNN 7.6.5 PyTorch v1.5.0

Platform: The Lassen Supercomputer (#14 on Top500.org) – 4 NVIDIA Volta GPUs per node connected with NVLink, CUDA 10.2
PyTorch at Scale: Training ResNet-50 on 256 V100 GPUs

- Training performance for 256 V100 GPUs on LLNL Lassen
  - ~10,000 Images/sec faster than NCCL training!

<table>
<thead>
<tr>
<th>Distributed Framework</th>
<th>Torch.distributed</th>
<th>Horovod</th>
<th>DeepSpeed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Images/sec on 256 GPUs</td>
<td>61,794</td>
<td>72,120</td>
<td>84,659</td>
</tr>
<tr>
<td>Communication Backend</td>
<td>NCCL</td>
<td>MVAPICH2-GDR</td>
<td>NCCL</td>
</tr>
</tbody>
</table>
Accelerating DL and ML Applications with MVAPICH2 MPI Libraries

- MPI-driven Deep Learning
  - CPU-based Deep Learning
  - GPU-based Deep Learning
- Out-of-core DNN training
- Exploiting Hybrid (Data and Model) Parallelism
- Use-Case: AI-Driven Digital Pathology
- Commercial Support and Products
HyPar-Flow: Hybrid and Model Parallelism for CPUs

- Data-Parallelism—only for models that fit the memory
- Out-of-core models
  - Deeper model $\rightarrow$ Better accuracy but more memory required!
- *Model parallelism can work for out-of-core models!*
- Key Challenges
  - Model Partitioning is difficult for application programmers
  - Finding the right partition (grain) size is hard
    - *cut at which layer and why?*
  - Developing a practical system for model-parallelism
    - Redesign DL Framework or create additional layers?
    - Existing Communication middleware or extensions needed?

---

Out-of-core Training with HyPar-Flow (512 nodes on TACC Frontera)

- ResNet-1001 with variable batch size
- Approach:
  - 48 model-partitions for 56 cores
  - 512 model-replicas for 512 nodes
  - Total cores: $48 \times 512 = 24,576$
- Speedup
  - **253X** on 256 nodes
  - **481X** on 512 nodes
- Scaling Efficiency
  - **98%** up to 256 nodes
  - **93.9%** for 512 nodes

Exploiting Model Parallelism in AI-Driven Digital Pathology

- Pathology whole slide image (WSI)
  - Each WSI = 100,000 x 100,000 pixels
  - Can not fit in a single GPU memory
  - Tiles are extracted to make training possible

- Two main problems with tiles
  - Restricted tile size because of GPU memory limitation
  - Smaller tiles loose structural information

- Can we use Model Parallelism to train on larger tiles to get better accuracy and diagnosis?

- Reduced training time significantly on OpenPOWER + NVIDIA V100 GPUs
  - 32 hours (1 node, 1 GPU) -> 7.25 hours (1 node, 4 GPUs) -> 27 mins (32 nodes, 128 GPUs)


Accelerating DL and ML Applications with MVAPICH2 MPI Libraries

- MPI-driven Deep Learning
  - CPU-based Deep Learning
  - GPU-based Deep Learning
- Out-of-core DNN training
- Exploiting Hybrid (Data and Model) Parallelism
- Use-Case: AI-Driven Digital Pathology
- Commercial Support and Products
Commercial Support for MVAPICH2, HiBD, and HiDL Libraries

- Supported through X-ScaleSolutions (http://x-scalesolutions.com)
- Benefits:
  - Help and guidance with installation of the library
  - Platform-specific optimizations and tuning
  - Timely support for operational issues encountered with the library
  - Web portal interface to submit issues and tracking their progress
  - Advanced debugging techniques
  - Application-specific optimizations and tuning
  - Obtaining guidelines on best practices
  - Periodic information on major fixes and updates
  - Information on major releases
  - Help with upgrading to the latest release
  - Flexible Service Level Agreements
- Support being provided to National Laboratories and International Supercomputing Centers
Silver ISV Member for the OpenPOWER Consortium

- Silver ISV member of the OpenPOWER Consortium
- Provides flexibility:
  - To have MVAPICH2, HiDL and HiBD libraries getting integrated into the OpenPOWER software stack
  - A part of the OpenPOWER ecosystem
  - Can participate with different vendors for bidding, installation and deployment process
- Introduced two new integrated products with support for OpenPOWER systems (Presented at the 2019 OpenPOWER North America Summit)
  - X-ScaleHPC
  - X-ScaleAI
X-ScaleHPC Package

• Scalable solutions of communication middleware based on OSU MVAPICH2 libraries

• “out-of-the-box” fine-tuned and optimal performance on various HPC systems including CPUs and GPUs

• MPI communication offloading capabilities to ARM based smart NICs (such as Mellanox Bluefield NICs)

Please refer to the presentation made at the 8th Annual MVAPICH User group Meeting (MUG) in Aug ‘20

http://mug.mvapich.cse.ohio-state.edu/program/
**X-ScaleAI Product and Features**

- **Aim:** High-performance solution for distributed training for your complex AI problems on modern HPC platforms

- **Features:**
  - Powered by MVAPICH2 libraries
  - Great performance and scalability as delivered by MVAPICH2 libraries
  - Integrated packaging to run various Deep Learning Frameworks (TensorFlow, PyTorch, MXNet, and others)
  - Targeted for both CPU-based and GPU-based Deep Learning Training
  - **Integrated profiling and introspection support for Deep Learning Applications across the stacks (DeepIntrospect)**
    - Provides cross-stack performance analysis in a visual manner and help users to optimize their DL applications and harness higher performance and scalability
  - **Out-of-the-box optimal performance**
    - Tuned for various CPU- and GPU-based HPC systems
  - **One-click deployment and execution**
    - Do not need to struggle for many hours
  - Support for x86 and OpenPOWER platforms
  - Support for InfiniBand, RoCE and NVLink Interconnects
X-ScaleAI Product with DeepIntrospect (DI) Capability

More details in MUG ‘20 talk, Contact for a demo and free-trial! (contactus@x-scalesolutions.com)
Conclusions

- HPC and DL applications require high-performance and scalability
- Requires high-performance middleware designs while exploiting modern interconnects
- Provided a set of solutions to achieve
  - MPI (MVAPICH2)-driven solution for HPC applications
  - MPI (MVAPICH2)-driven solution for DL applications with TensorFlow, PyTorch frameworks
  - Out-of-core training and Hybrid Parallelism
  - Commercial support and product with Deep Introspection Capability
- Will continue to enable the HPC and DL community to achieve scalability and high-performance with OpenPOWER Platforms
Funding Acknowledgments

Funding Support by

Equipment Support by
Acknowledgments to all the Heroes (Past/Current Students and Staffs)

**Current Students (Graduate)**
- Q. Anthony (Ph.D.)
- M. Bayatpour (Ph.D.)
- C.-C. Chun (Ph.D.)
- A. Jain (Ph.D.)
- M. Kedia (M.S.)
- K. S. Khorassani (Ph.D.)
- P. Kousha (Ph.D.)
- N. S. Kumar (M.S.)
- B. Ramesh (Ph.D.)
- K. K. Suresh (Ph.D.)
- N. Sarkauskas (Ph.D.)
- S. Srivastava (M.S.)
- S. Xu (Ph.D.)
- Q. Zhou (Ph.D.)

**Past Students**
- A. Awan (Ph.D.)
- 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.)
- S. Chakraborty (Ph.D.)
- N. Dandapanthula (M.S.)
- V. Dhanraj (M.S.)
- C.-H. Chu (Ph.D.)
- T. Gangadharappa (M.S.)
- K. Gopalakrishnan (M.S.)
- J. Hashmi (Ph.D.)
- 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.)
- K. Raj (M.S.)
- R. Rajachandrasekar (Ph.D.)
- D. Shankar (Ph.D.)
- G. Santhanaraman (Ph.D.)
- S. Sarkauskas (B.S.)
- A. Singh (Ph.D.)
- J. Sridhar (M.S.)
- S. Sur (Ph.D.)
- H. Subramoni (Ph.D.)
- A. Vishnu (Ph.D.)
- J. Wu (Ph.D.)
- W. Yu (Ph.D.)
- J. Zhang (Ph.D.)
- J. Vienne
- H. Wang

**Past Post-Docs**
- D. Banerjee
- X. Besseron
- H.-W. Jin
- J. Lin
- M. Luo
- E. Mancini
- K. Manian
- S. Marcarelli
- A. Ruhela

**Past Programmers**
- D. Bureddy
- J. Perkins

**Past Research Scientists**
- M. Arnold

**Current Research Scientists**
- A. Shafi
- H. Subramoni

**Current Senior Research Associate**
- J. Hashmi

**Current Research Specialist**
- J. Smith

**Current Software Engineer**
- A. Reifsteck

**Current Post-docs**
- M. S. Ghazimeersaeed

**Past Research Scientists**
- K. Hamidouche
- S. Sur
- X. Lu

**Past Programmers**
- D. Bureddy
- J. Perkins

**Past Research Specialist**
- M. Arnold
Thank You!

panda@cse.ohio-state.edu

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

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

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

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