Exploiting Multi-core Processors for HPC and Deep Learning: The MVAPICH2 Approach

Intel Booth Talk at SC '19

by

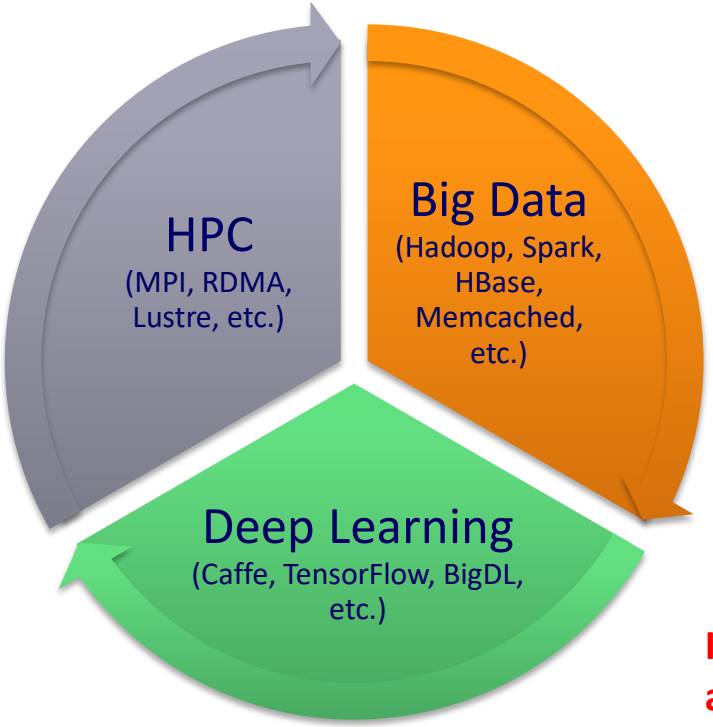
Dhabaleswar K. (DK) Panda

The Ohio State University

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

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

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!!

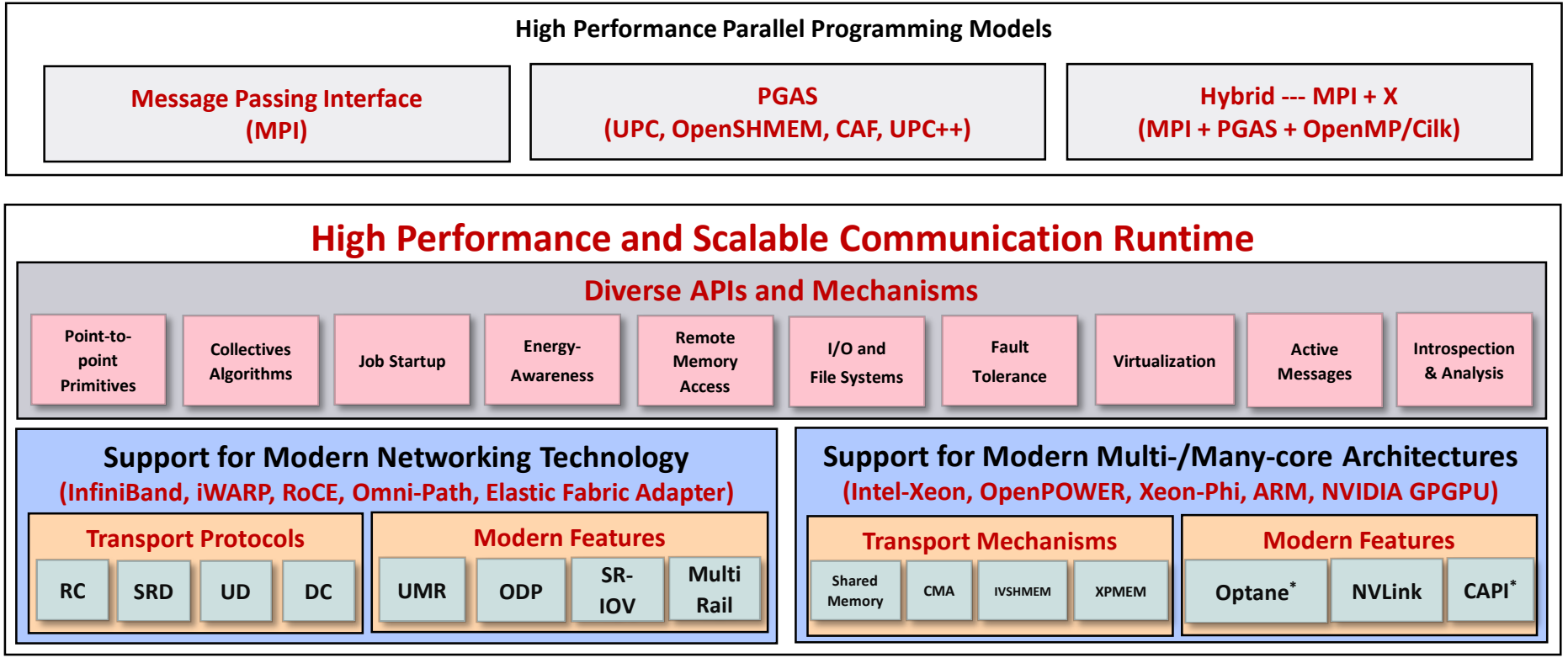
Overview of the MVAPICH2 Project

- 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 (Supercomputing '02)
  - 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 3,050 organizations in 89 countries**
  - **More than 615,000 (> 0.6 million) downloads from the OSU site directly**
  - Empowering many TOP500 clusters (Nov '19 ranking)
    - 3rd, 10,649,600-core (Sunway TaihuLight) at National Supercomputing Center in Wuxi, China
    - 5th, 448, 448 cores (Frontera) at TACC
    - 8th, 391,680 cores (ABCI) in Japan
    - 14th, 570,020 cores (Nurion) in South Korea and many others
  - Available with software stacks of many vendors and Linux Distros (RedHat, SuSE, and OpenHPC)
  - **http://mvapich.cse.ohio-state.edu**

Partner in the #5th TACC Frontera System

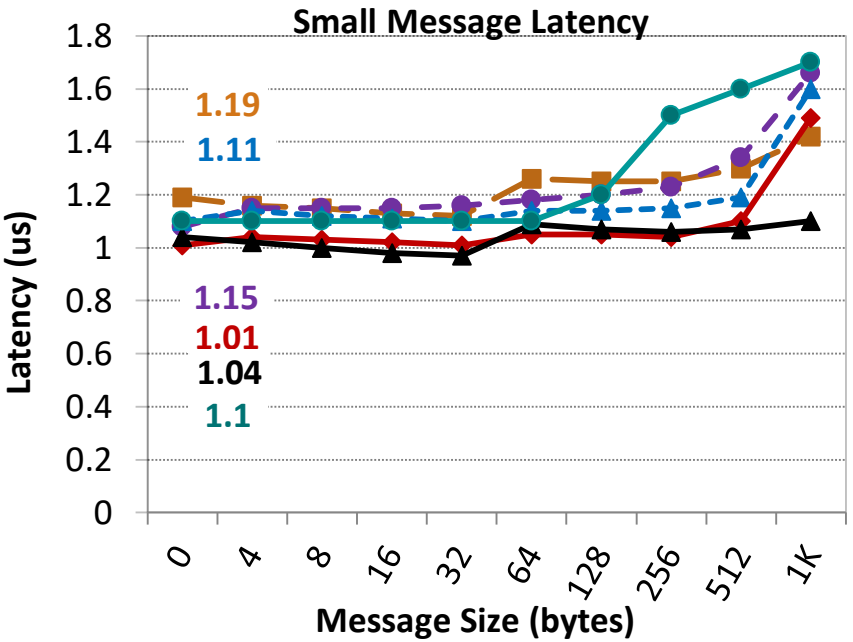
- Empowering Top500 systems for over a decade

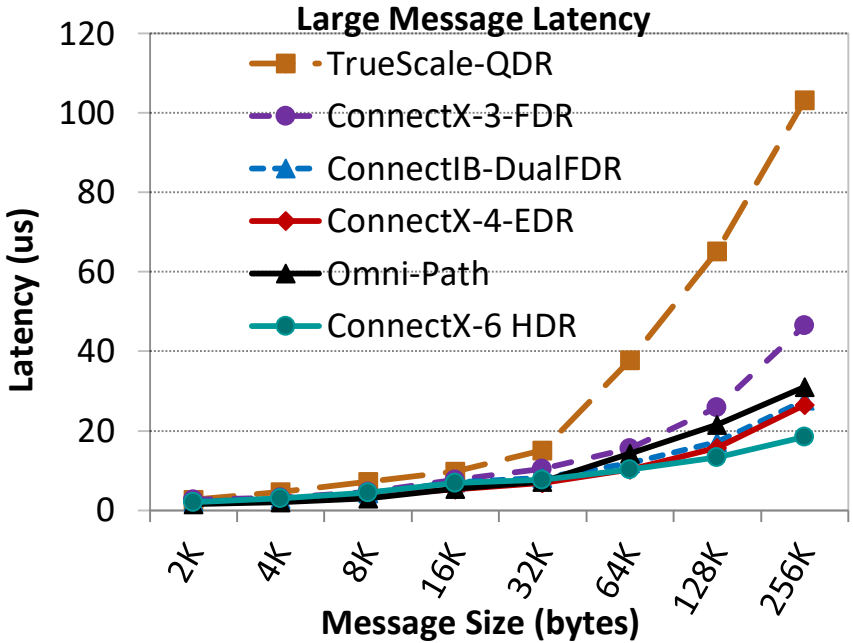
Architecture of MVAPICH2 Software Family (HPC and DL)



* Upcoming

One-way Latency: MPI over IB with MVAPICH2





TrueScale-QDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB switch

ConnectX-3-FDR - 2.8 GHz Deca-core (IvyBridge) Intel PCI Gen3 with IB switch

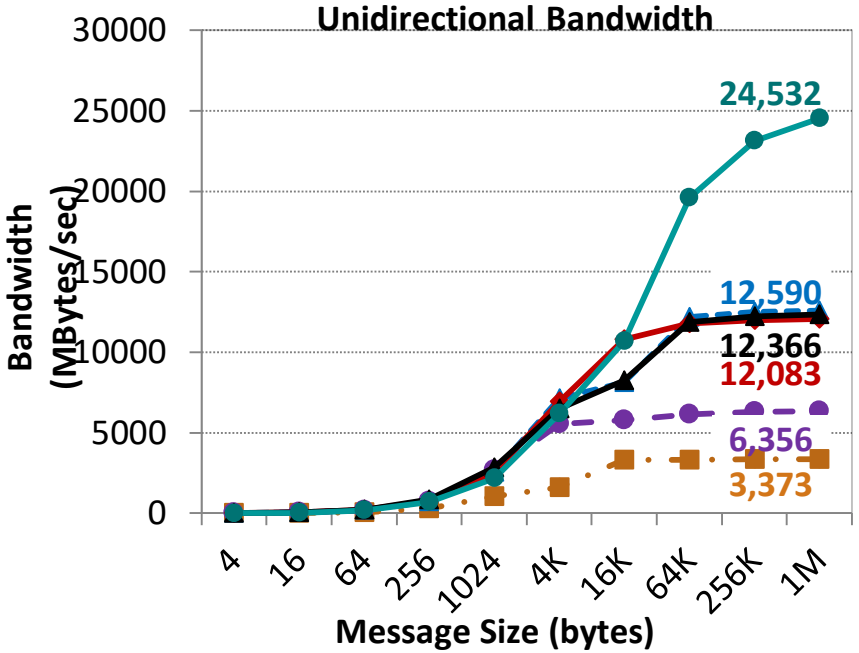
ConnectIB-Dual FDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB switch

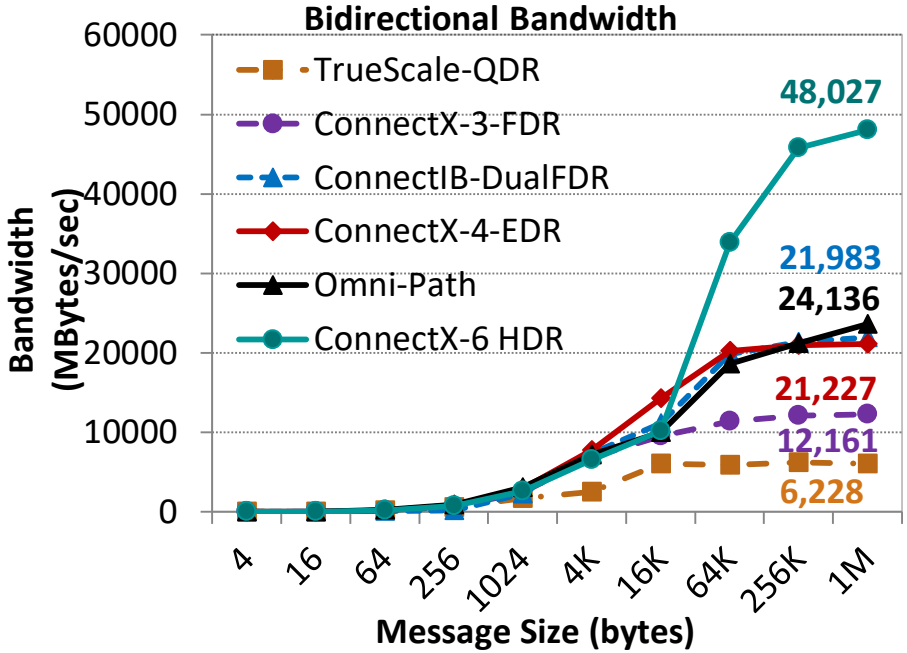
ConnectX-4-EDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB Switch

Omni-Path - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with Omni-Path switch

ConnectX-6-HDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB Switch

Bandwidth: MPI over IB with MVAPICH2





TrueScale-QDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB switch

ConnectX-3-FDR - 2.8 GHz Deca-core (IvyBridge) Intel PCI Gen3 with IB switch

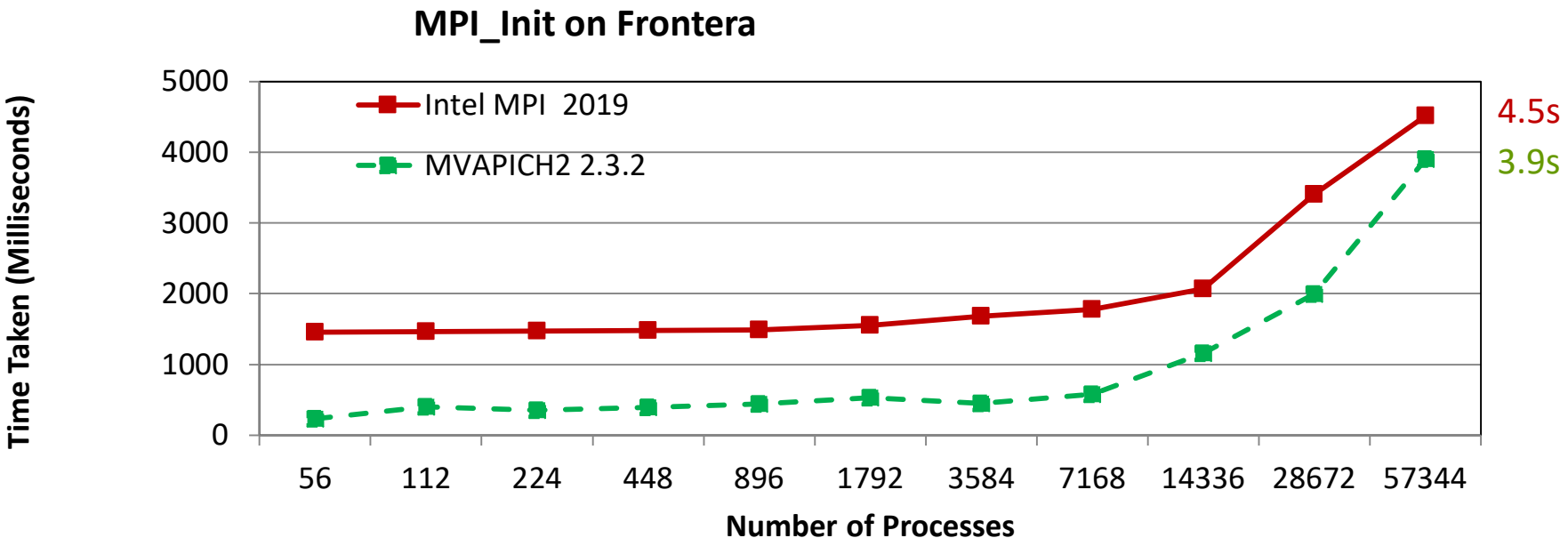
ConnectIB-Dual FDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB switch

ConnectX-4-EDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB Switch

Omni-Path - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with Omni-Path switch

ConnectX-6-HDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB Switch

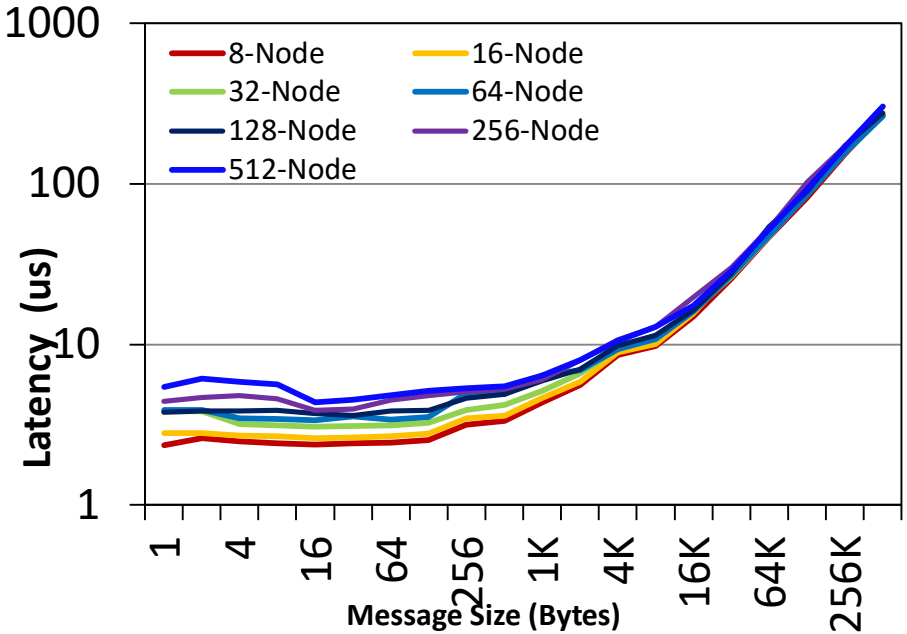
Startup Performance on TACC Frontera

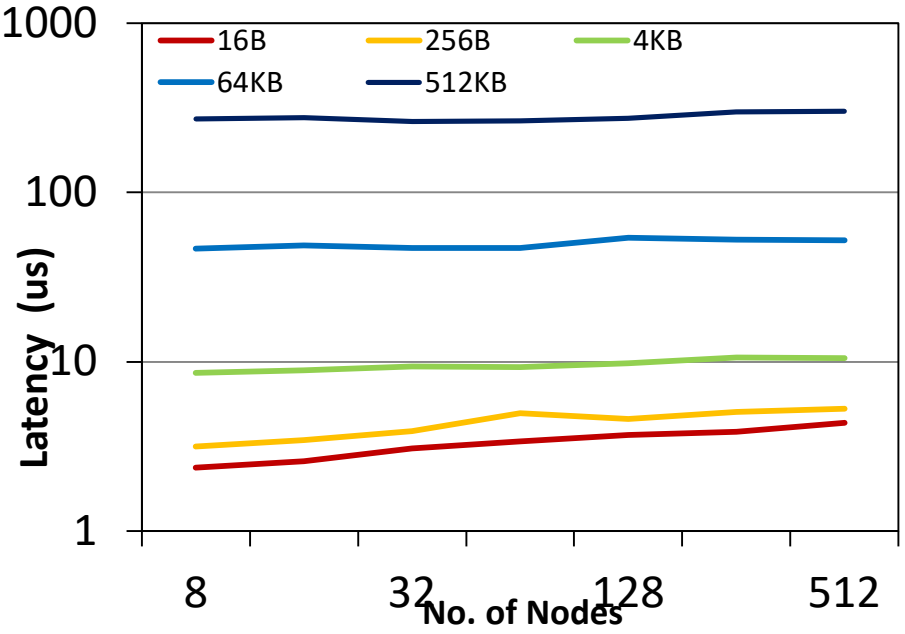


- MPI_Init takes 3.9 seconds on 57,344 processes on 1,024 nodes
- All numbers reported with 56 processes per node

New designs available in MVAPICH2-2.3.2

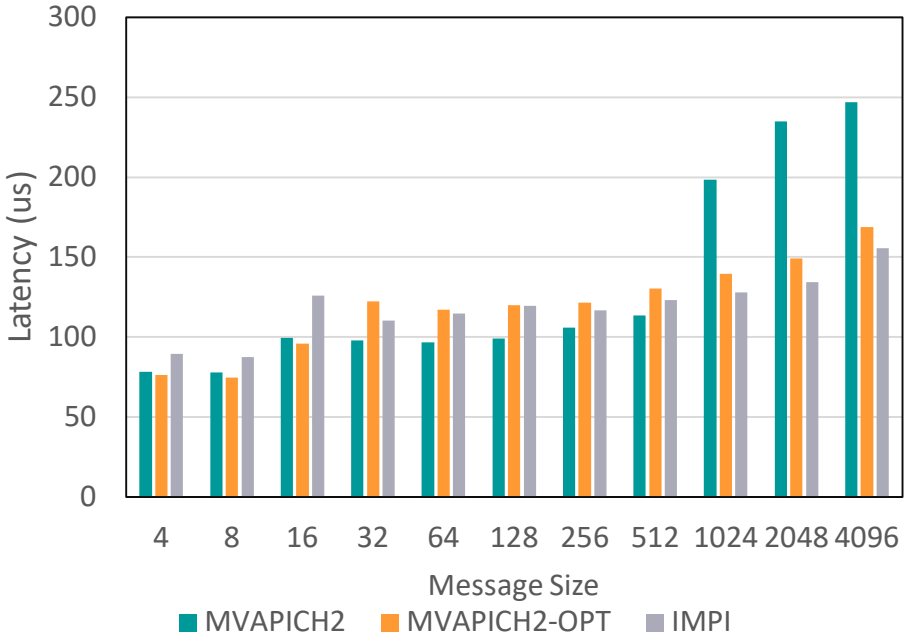
Multicast-based Bcast on Frontera

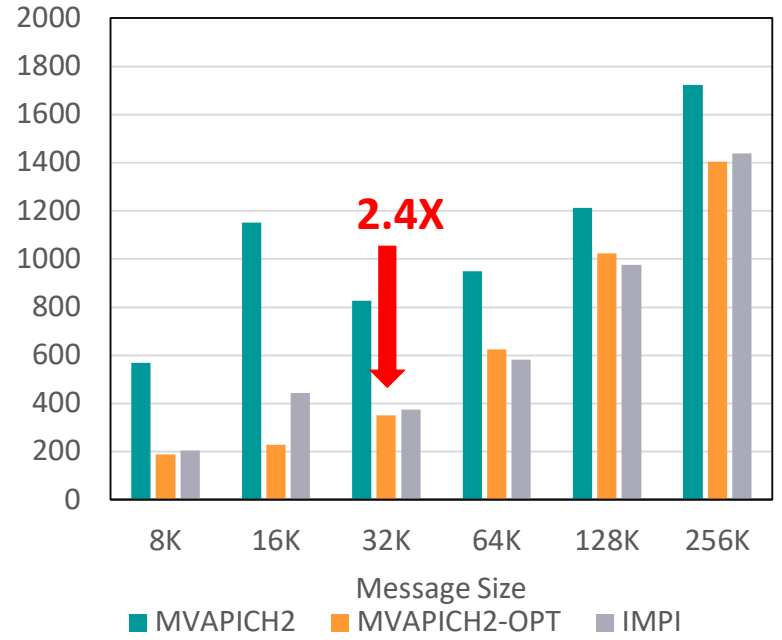




- Scalable broadcast with increase in system size

MPI_Allreduce on KNL + Omni-Path (10,240 Processes)





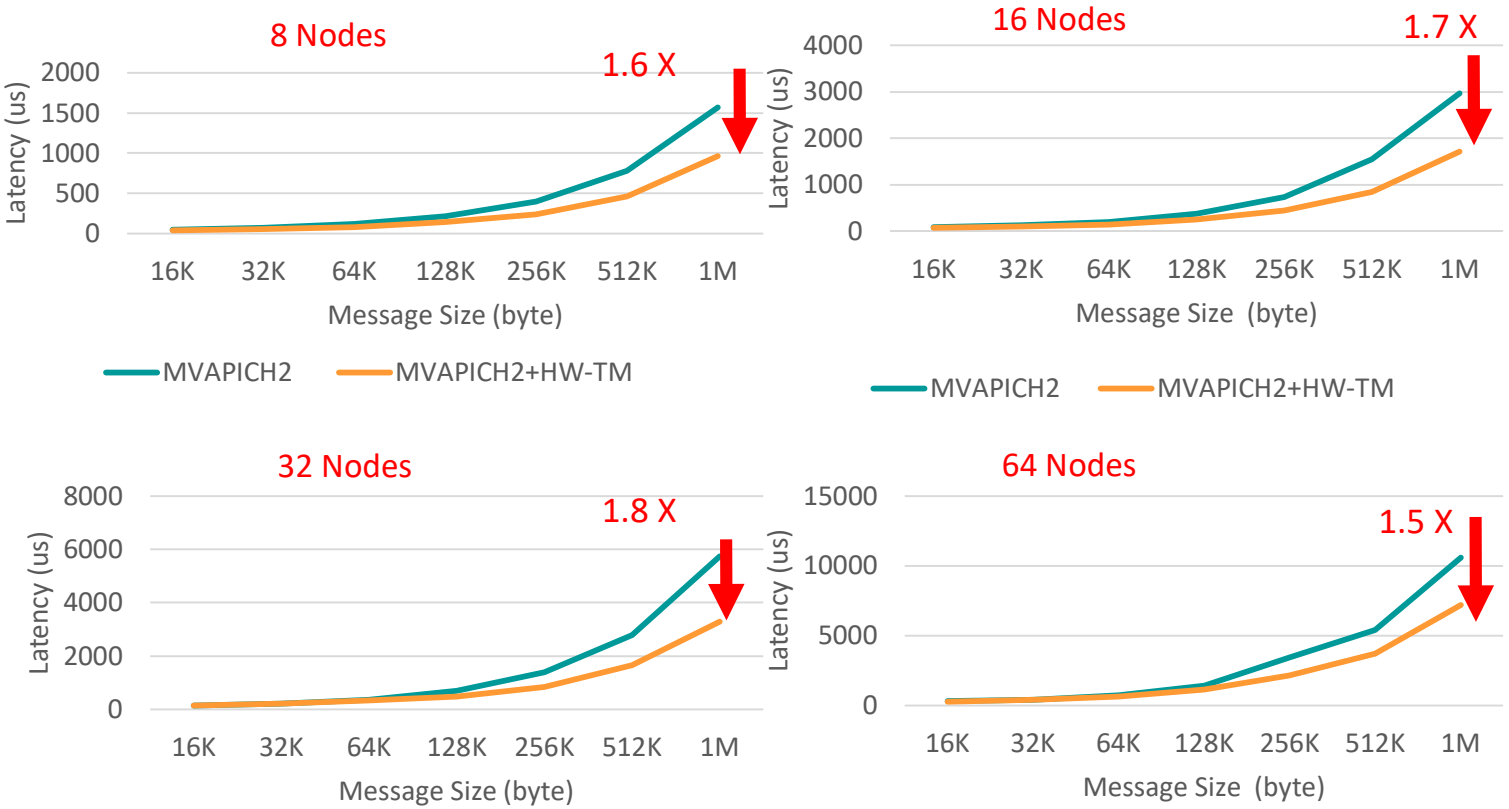
OSU Micro Benchmark 64 PPN

- For MPI_Allreduce latency with 32K bytes, MVAPICH2-OPT can reduce the latency by 2.4X

M. Bayatpour, S. Chakraborty, H. Subramoni, X. Lu, and D. K. Panda, Scalable Reduction Collectives with Data Partitioning-based Multi-Leader Design, SuperComputing '17.

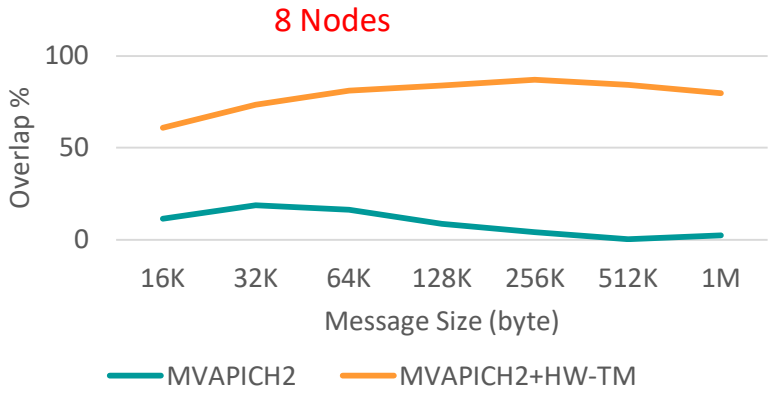
Available since MVAPICH2-X 2.3b

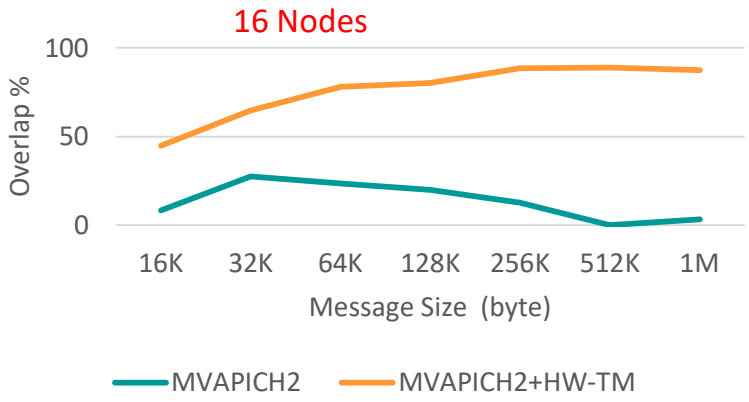
Performance of MPI_Ialltoall using HW Tag Matching on Frontera

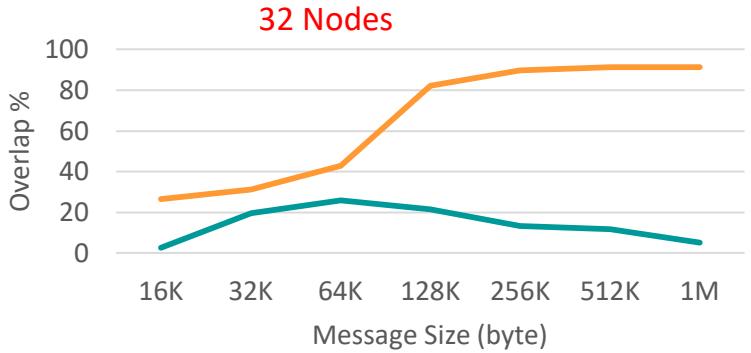


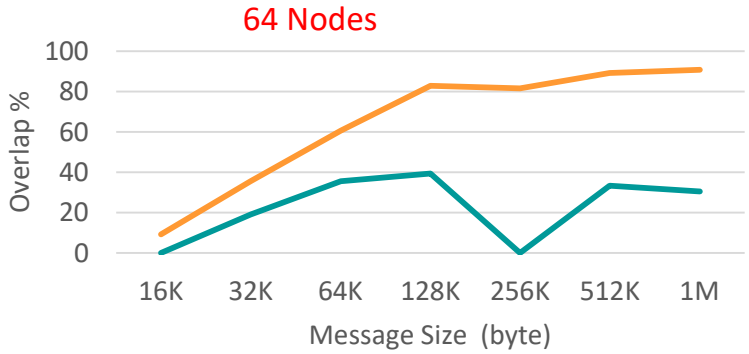
- Up to 1.8x Performance Improvement
- Sustained benefits as system size increases

Overlap with MPI_Ialltoall using HW Tag Matching on Frontera



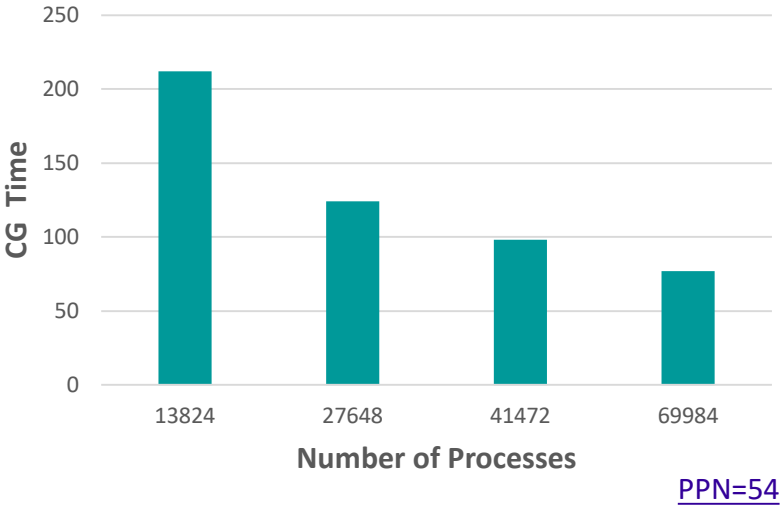






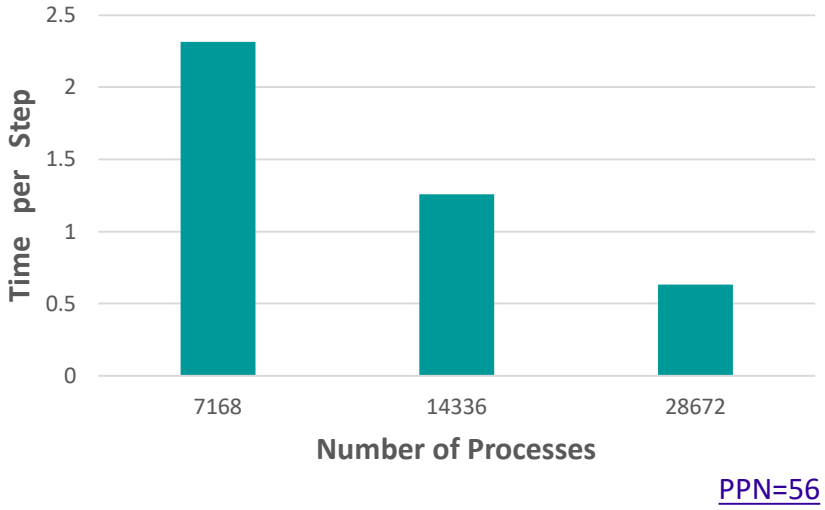
- Maximizing the overlap of communication and computation
- Sustained benefits as system size increases

Evaluation of Applications on Frontera (Cascade Lake + HDR100)



PPN=54

MIMD Lattice Computation (MILC)



PPN=56

WRF2

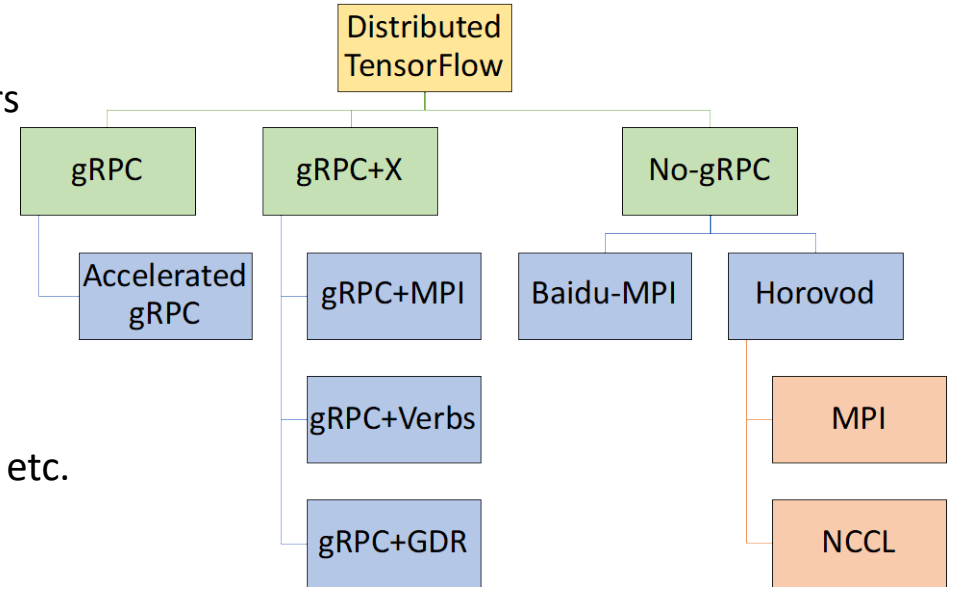
Performance of MILC and WRF2 applications scales well with increase in system size

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?

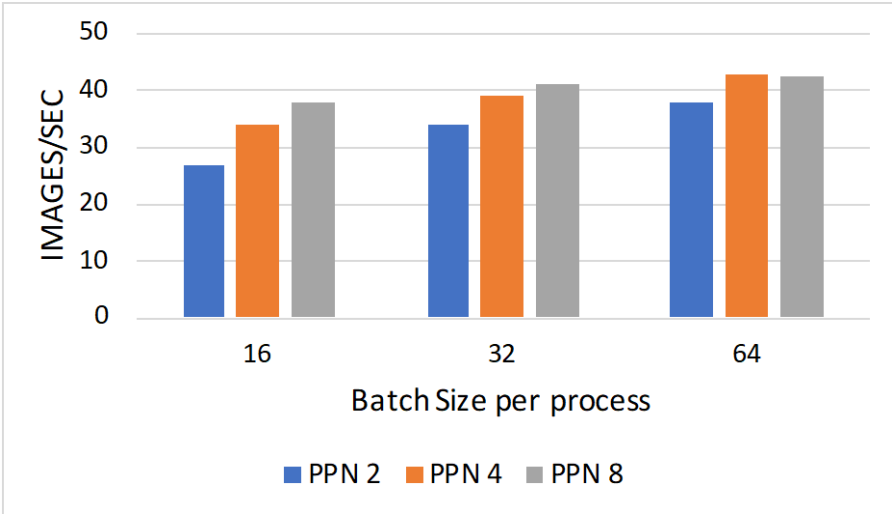
Data Parallel Training with TensorFlow

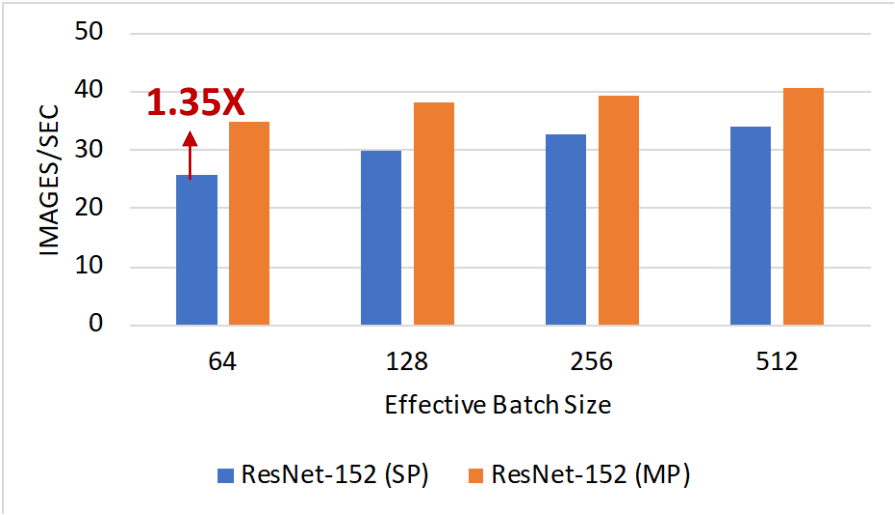
- gRPC
  - Officially available and supported
  - Open-source – can be enhanced by others
  - Accelerated gRPC (add RDMA to gRPC)
- gRPC+X
  - Use gRPC for bootstrap and rendezvous
  - ***Actual communication is in "X"***
  - X→ MPI, Verbs, GPUDirect RDMA (GDR), etc.
- No-gRPC
  - Baidu – the first one to use MPI Collectives for TF
  - Horovod – Use NCCL, or MPI, or any other future library (e.g. IBM DDL support recently added)



*Awan et al., "Scalable Distributed DNN Training using TensorFlow and CUDA-Aware MPI: Characterization, Designs, and Performance Evaluation ", CCGrid '19.

CPU-based Training: Single-Node Multi-Process (MP) mode





ResNet-152 Training performance

- **BS=64, 4ppn is better; BS=32, 8ppn is slightly better**
- **However, keeping effective batch size (EBS) low is more important! – Why? (DNN does not converge to SOTA when batch size is large)**

ResNet-152: Single Process (SP) vs. Multi-Process(MP)

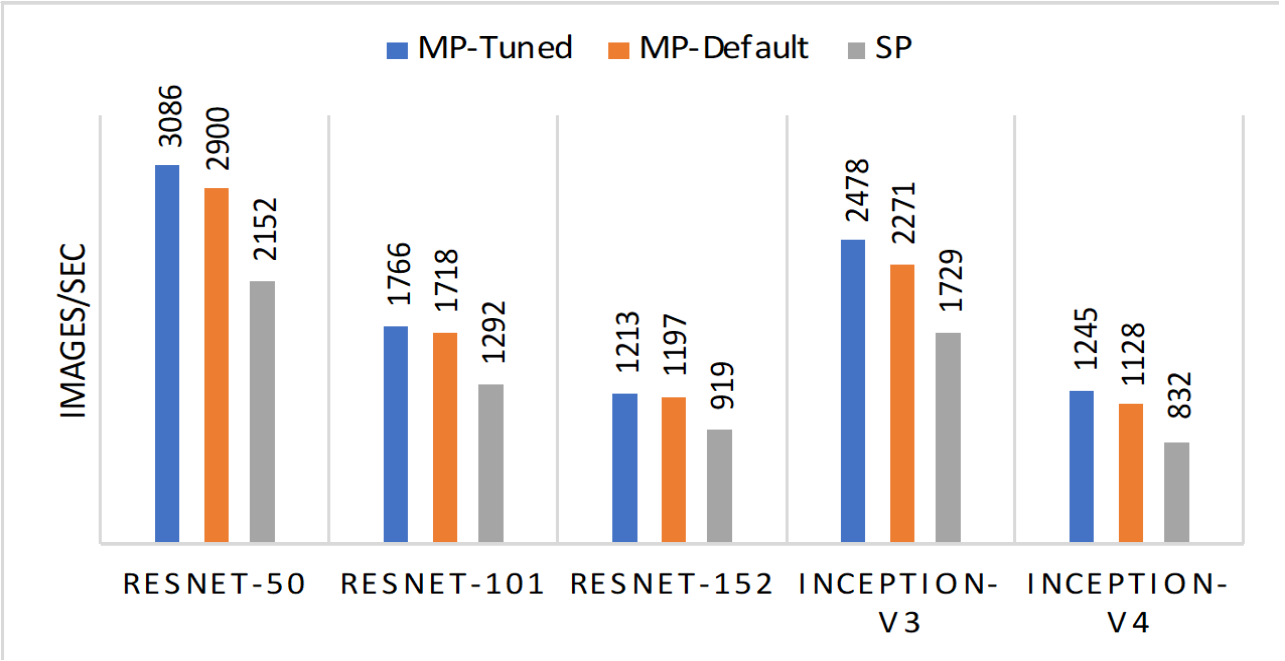
- **MP is better for all effective batch sizes**
- **Up to 1.35X better performance for MP compared to SP for BS=64.**

*Jain et al., "Performance Characterization of DNN Training using TensorFlow and PyTorch on Modern Clusters", Cluster '19.

CPU-based Training: Multi-Process (MN): MP vs. SP?

Skylake-3 (48 cores, 96 threads)

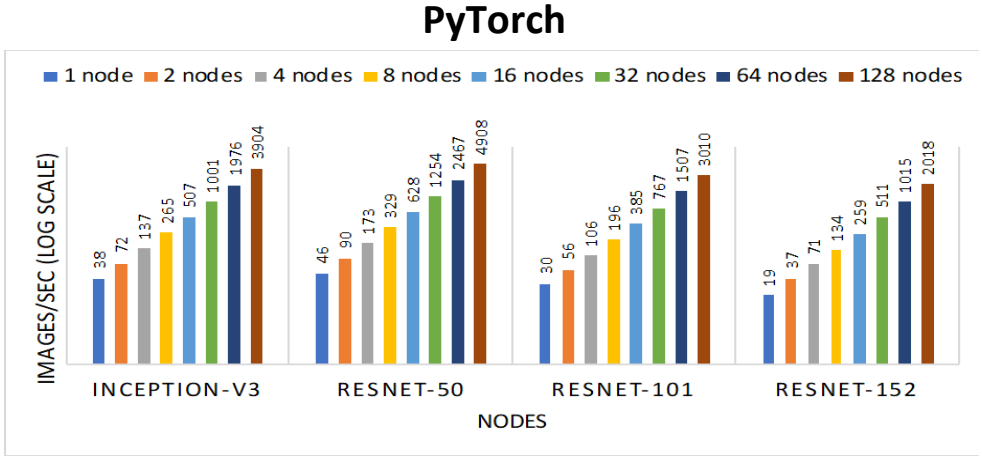
- **Scale—32 nodes**
- **MP-Tuned—up to 1.5X better than SP**
- **MP-Tuned—10% better than MP-Default**
- **Why MP-Tuned is better?**
  - **Uses the best possible number of inter-op and intra-op threads**

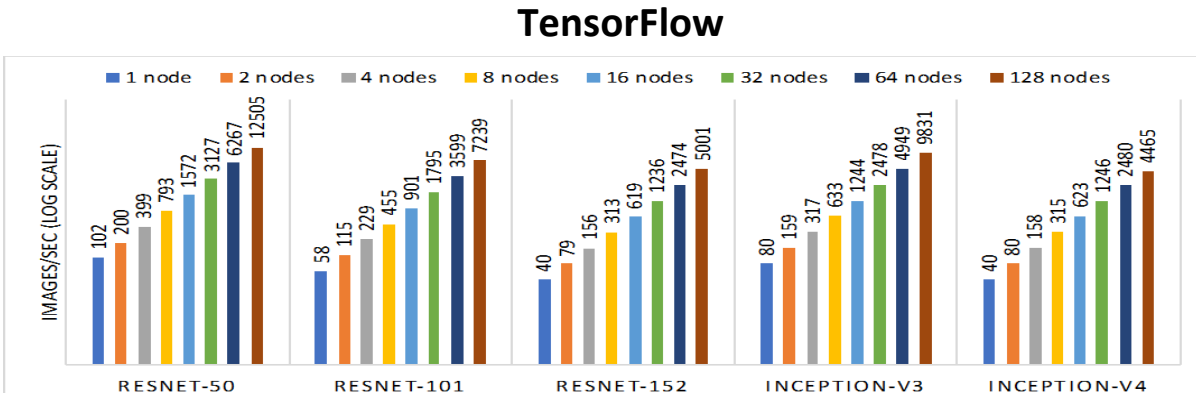


*Jain et al., "Performance Characterization of DNN Training using TensorFlow and PyTorch on Modern Clusters", Cluster '19.

Multi-Node Multi-Process (MN): TensorFlow vs. PyTorch

- This is an early experience with PyTorch
- TensorFlow is up to 2.5X faster than PyTorch for 128 Nodes.
- TensorFlow: up to 125X speedup for ResNet-152 on 128 nodes
- PyTorch: Scales well but overall lower performance than TensorFlow

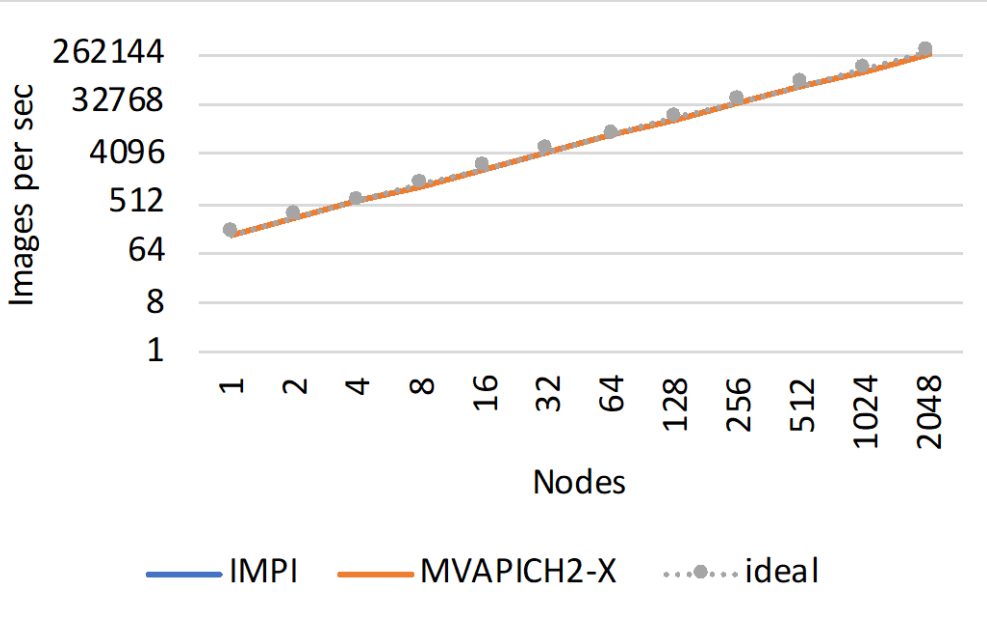




*Jain et al., "Performance Characterization of DNN Training using TensorFlow and PyTorch on Modern Clusters", Cluster '19.

Scaling ResNet-50 on TACC Frontera: 2,048 nodes!

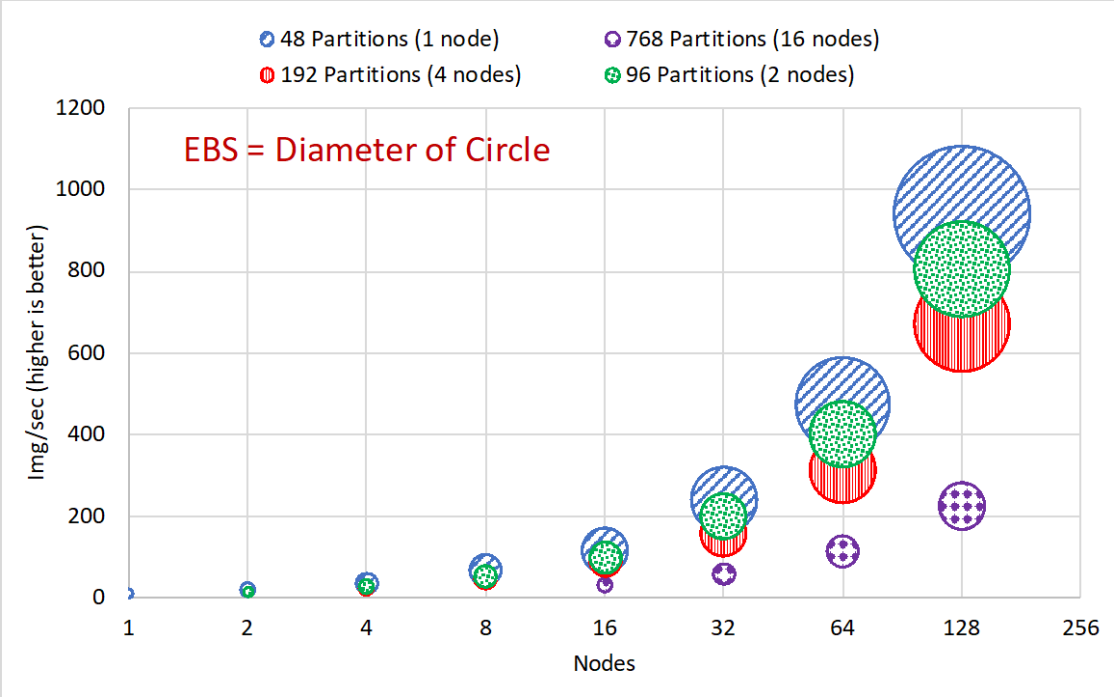
- Scaled TensorFlow to 2048 nodes on Frontera using MVAPICH2 and IntelMPI
- MVAPICH2 and IntelMPI give similar performance for DNN training
- Report a peak of 260,000 images/sec on 2048 nodes
- On 2048 nodes, ResNet-50 can be trained in 7 minutes!



*Jain et al., "Scaling TensorFlow, PyTorch, and MXNet using MVAPICH2 for High-Performance Deep Learning on Frontera", DLS '19 (in conjunction with SC '19).

Benchmarking HyPar-Flow on Stampede2

- CPU based Hybrid-Parallel (Data Parallelism and Model Parallelism) training on Stampede2
- Benchmark developed for various configuration
  - Batch sizes
  - No. of model partitions
  - No. of model replicas
- Evaluation on a very deep model
  - ResNet-1000 (a 1,000-layer model)



110x speedup on 128 Intel Xeon Skylake nodes (TACC Stampede2 Cluster)

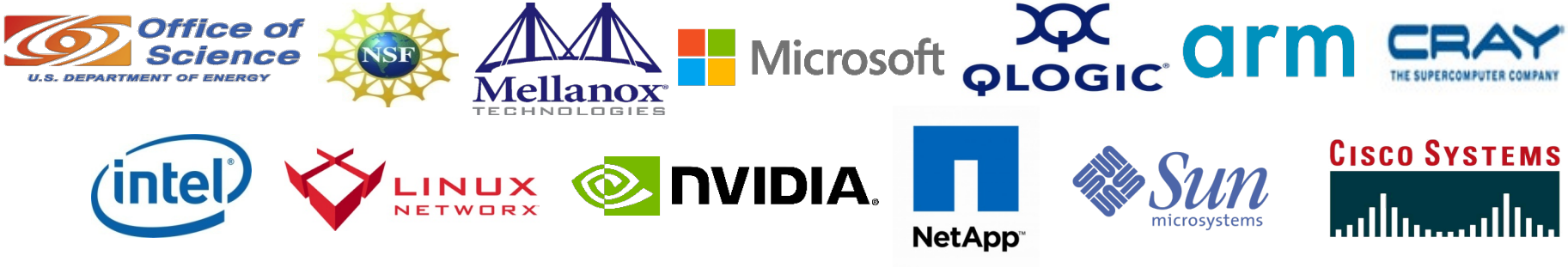
*Awan et al., "HyPar-Flow: Exploiting MPI and Keras for Hybrid Parallel Training of TensorFlow models", arXiv '19. https://arxiv.org/pdf/1911.05146.pdf

Conclusions

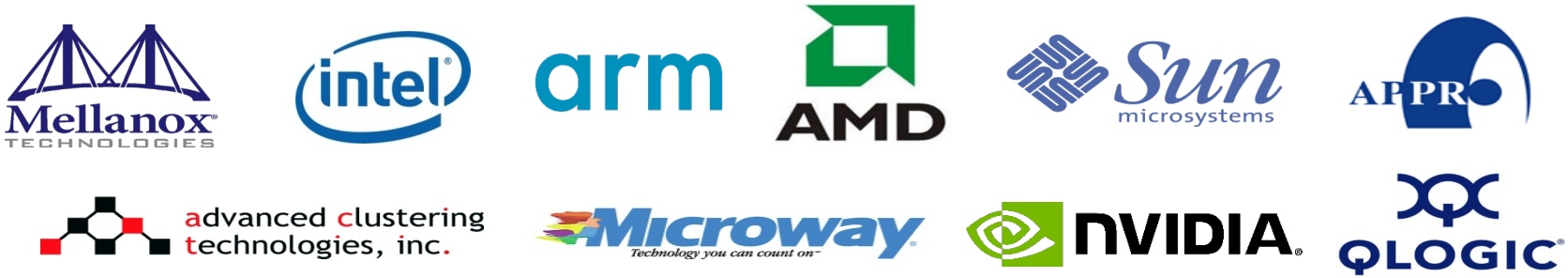
- Scalable performance for HPC and distributed training is getting important
- Requires high-performance middleware designs while exploiting modern interconnects
- Provided the approaches being taken care of by the MVAPICH2 project to achieve scalable distributed training
- Will continue to enable the HPC and DL communities to achieve scalability and high-performance for their scientific and distributed training workloads

Funding Acknowledgments

Funding Support by



Equipment Support by



Personnel Acknowledgments

Current Students (Graduate)

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

Current Research Scientist

- H. Subramoni

Current Post-doc

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

Current Students (Undergraduate)

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

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.)
- S. Chakraborthy (Ph.D.)
- 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.)
- D. Shankar (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

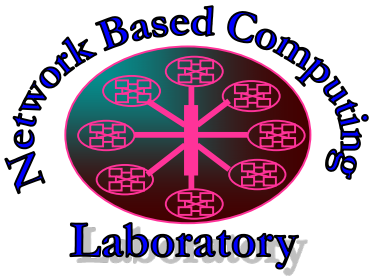
- M. Arnold

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
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/