Exploiting Emerging Multi-core Processors for HPC and Deep Learning using MVAPICH2 MPI Library

Talk at IXPUG '19

by

Dhabaleswar K. (DK) Panda

The Ohio State University

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

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

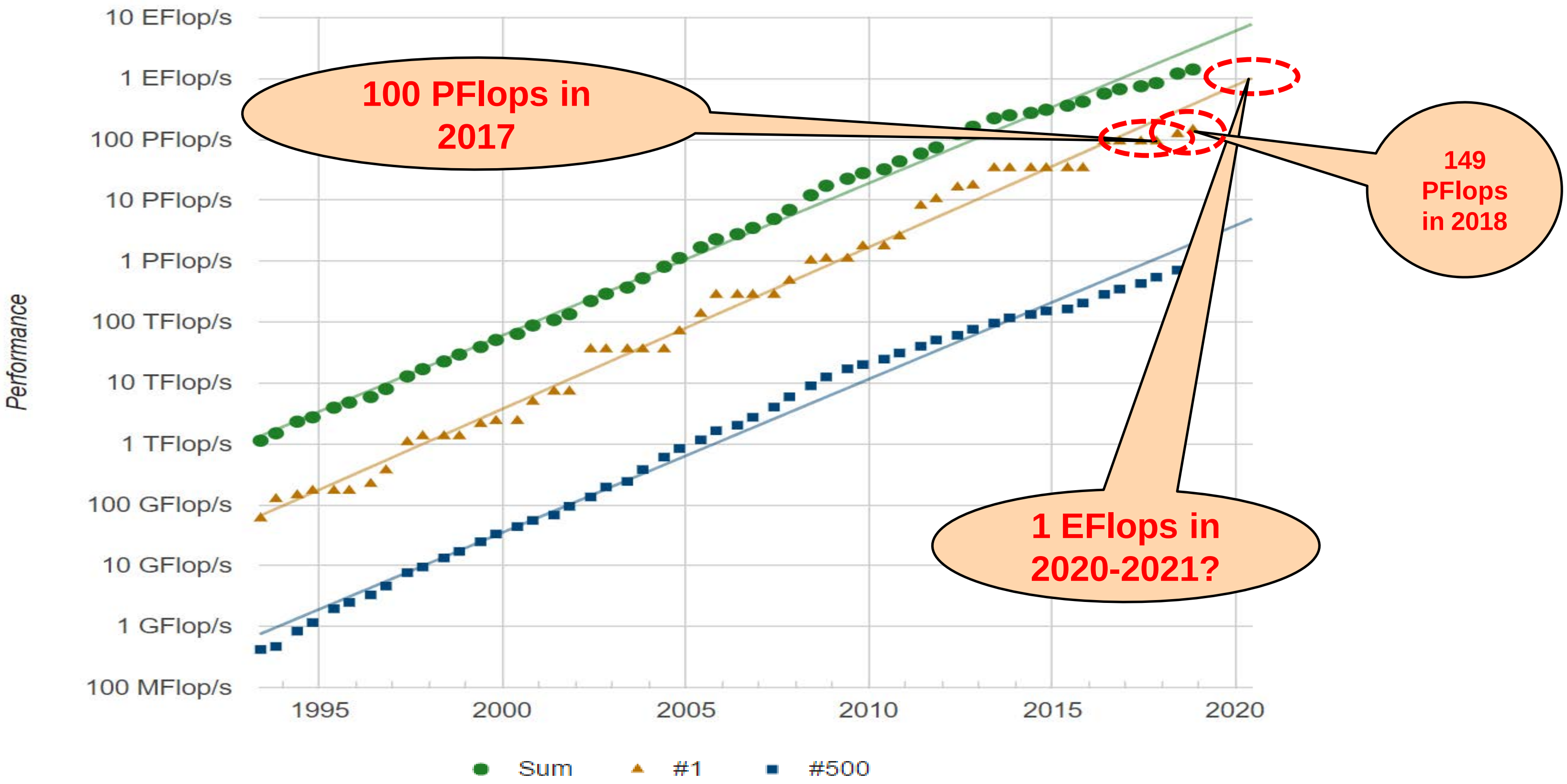
Hari Subramoni

The Ohio State University

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

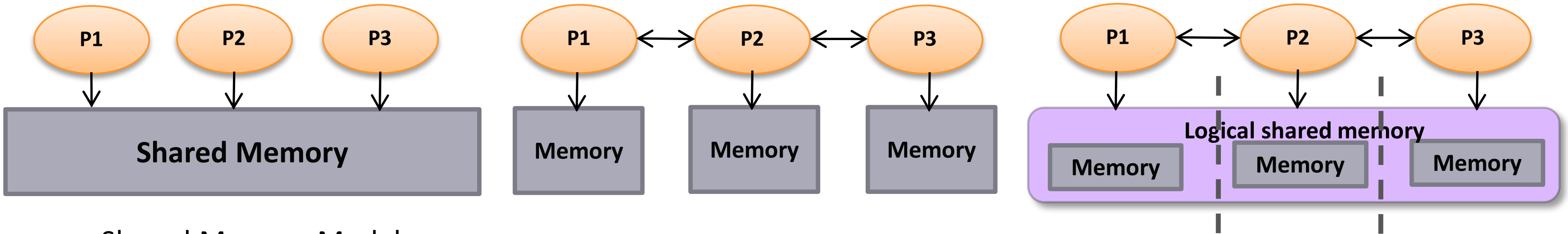
http://www.cse.ohio-state.edu/~subramon

High-End Computing (HEC): PetaFlop to ExaFlop



Expected to have an ExaFlop system in 2020-2021!

Parallel Programming Models Overview



Shared Memory Model

SHMEM, DSM

Distributed Memory Model

MPI (Message Passing Interface)

Partitioned Global Address Space (PGAS)

OpenSHMEM, UPC, Chapel, X10, CAF, …

- Programming models provide abstract machine models
- Models can be mapped on different types of systems
  - e.g. Distributed Shared Memory (DSM), MPI within a node, etc.
- PGAS models and Hybrid MPI+PGAS models are gradually receiving importance

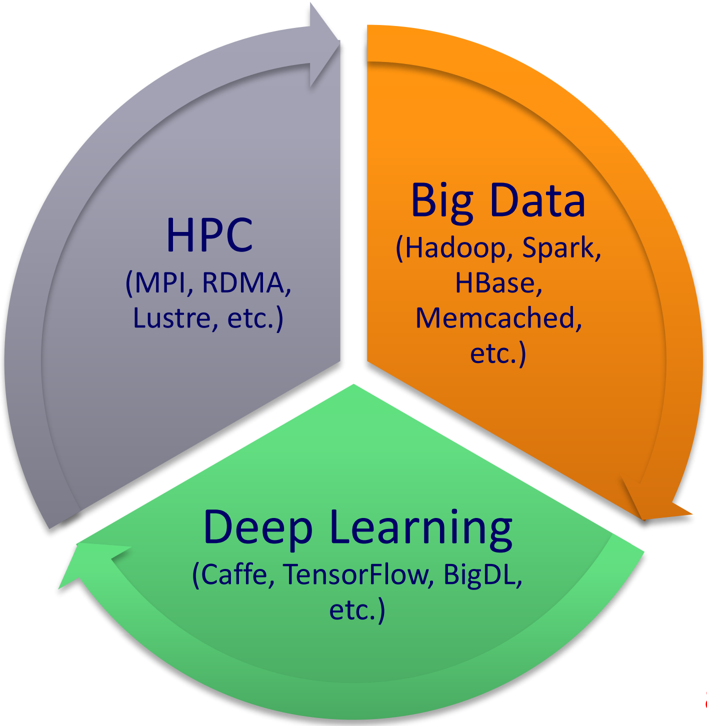
Broad Challenges in Designing Runtimes for (MPI+X) at Exascale

- Scalability for million to billion processors
  - Support for highly-efficient inter-node and intra-node communication (both two-sided and one-sided)
  - Scalable job start-up
  - Low memory footprint
- Scalable Collective communication
  - Offload
  - Non-blocking
  - Topology-aware
- Balancing intra-node and inter-node communication for next generation nodes (128-1024 cores)
  - Multiple end-points per node
- Support for efficient multi-threading
- Integrated Support for Accelerators (GPGPUs and FPGAs)
- Fault-tolerance/resiliency
- QoS support for communication and I/O
- Support for Hybrid MPI+PGAS programming (MPI + OpenMP, MPI + UPC, MPI + OpenSHMEM, MPI+UPC++, CAF, …)
- Virtualization
- Energy-Awareness

Additional Challenges for Designing Exascale Software Libraries

- **Extreme Low Memory Footprint**
  - Memory per core continues to decrease
- **D-L-A Framework**
  - **D**iscover
    - Overall network topology (fat-tree, 3D, …), Network topology for processes for a given job
    - Node architecture, Health of network and node
  - **L**earn
    - Impact on performance and scalability
    - Potential for failure
  - **A**dapt
    - Internal protocols and algorithms
    - Process mapping
    - Fault-tolerance solutions
  - Low overhead techniques while delivering performance, scalability and fault-tolerance

Increasing Usage of HPC, Big Data and Deep Learning

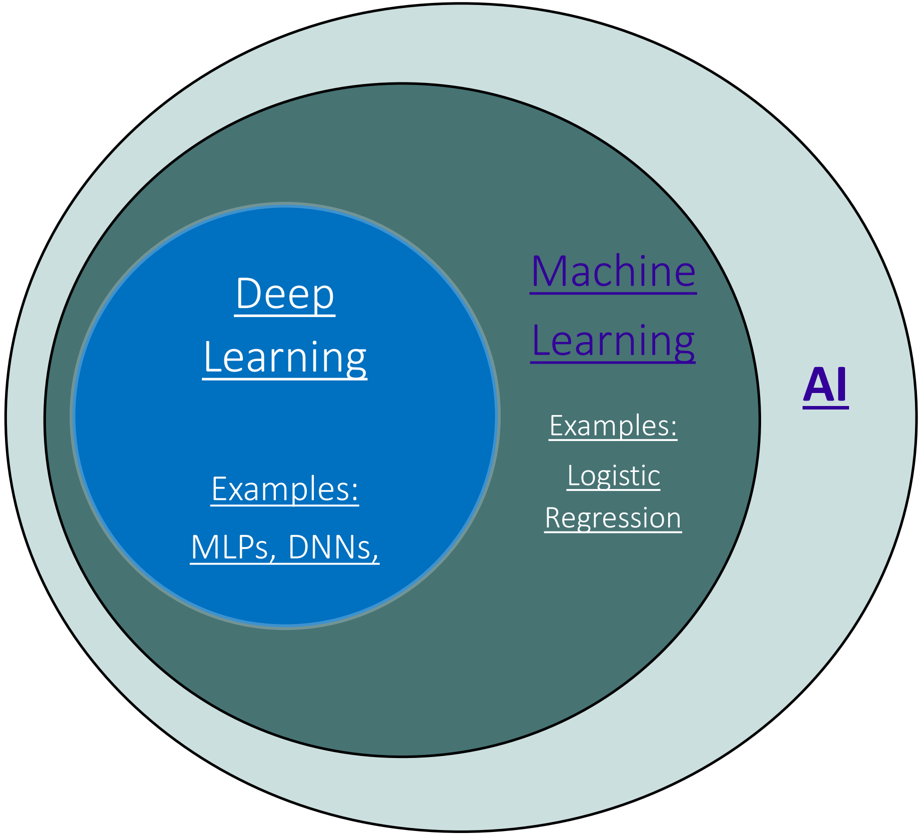


Convergence of HPC, Big Data, and Deep Learning!

Increasing Need to Run these applications on the Cloud!!

Understanding the Deep Learning Resurgence

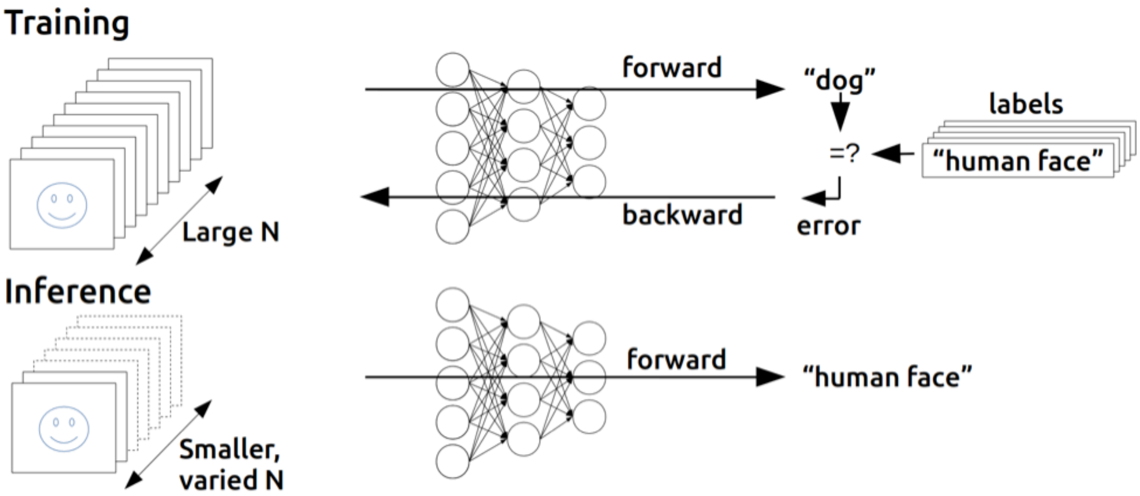
- Deep Learning (DL) is a sub-set of Machine Learning (ML)
  - Perhaps, the most revolutionary subset!
  - **Feature extraction** vs. **hand-crafted features**
- Deep Learning
  - A renewed interest and a lot of hype!
  - Key success: Deep Neural Networks (DNNs)
  - Everything was there since the late 80s except the **"computability of DNNs"** and **"diverse datasets"**



Adopted from: http://www.deeplearningbook.org/contents/intro.html

Key Phases of Deep Learning

- Training is compute intensive
  - Many passes over data
  - Can take days to weeks
  - Model adjustment is done
- Inference
  - Single pass over the data
  - Should take seconds
  - No model adjustment
- Challenge: How to make **"Training"** faster?
  - Need Parallel and Distributed Training…



Courtesy: https://devblogs.nvidia.com/

Broad Challenge:

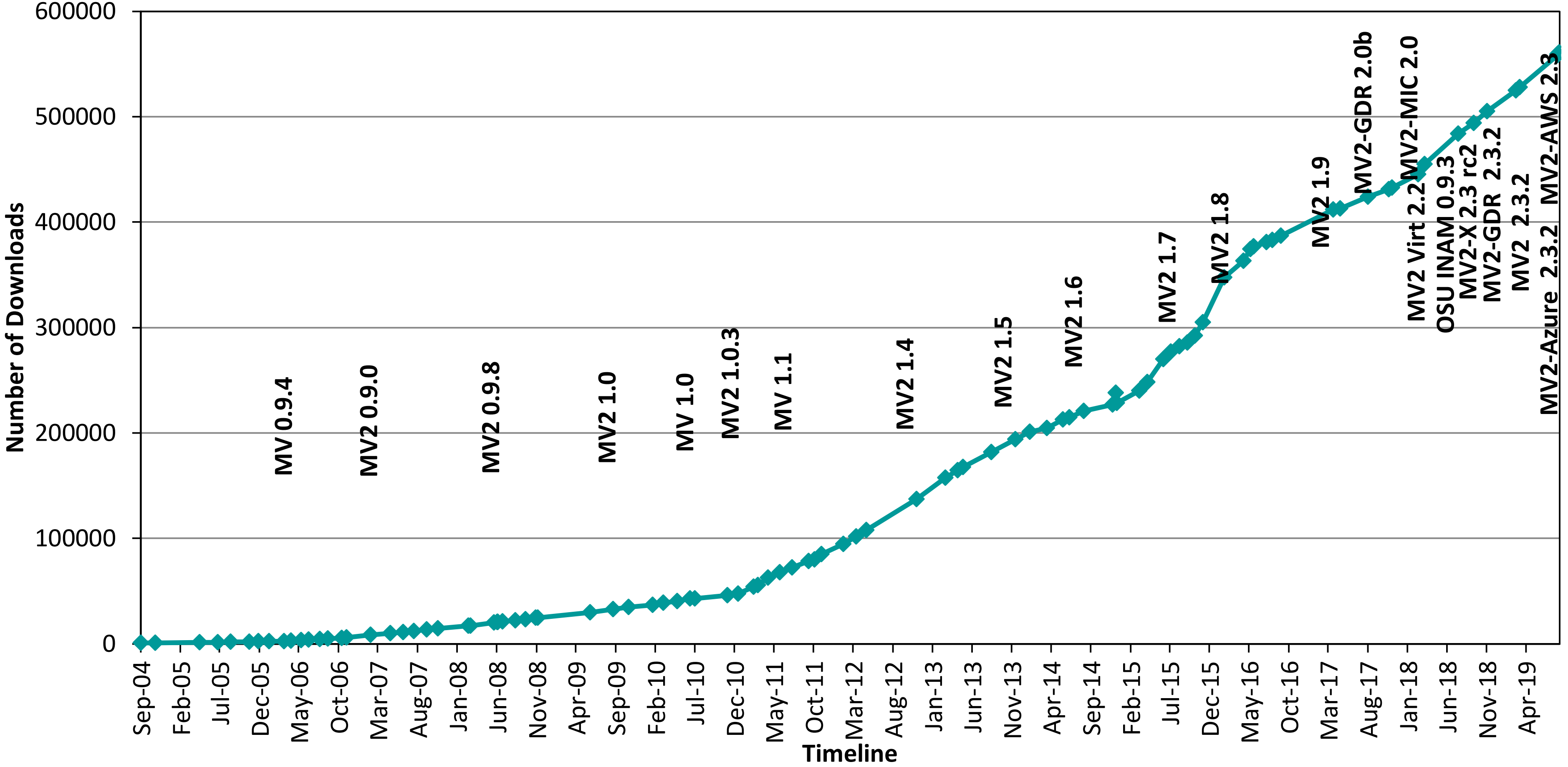
How to Design an Efficient MPI Library for Scalable HPC and Deep Learning (DL) by exploiting Multi-core Processors?

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 3,025 organizations in 89 countries**
  - **More than 589,000 (> 0.5 million) downloads from the OSU site directly**
  - Empowering many TOP500 clusters (Nov '18 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
    - 15th, 570,020 cores (Neurion) 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**
- Empowering Top500 systems for over a decade

Partner in the TACC Frontera System

MVAPICH2 Release Timeline and Downloads



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, 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

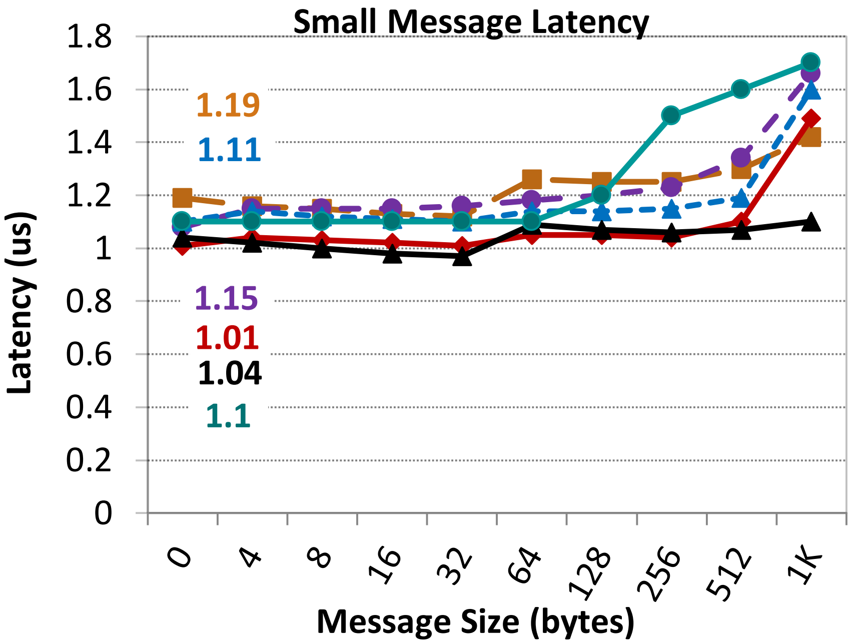
MVAPICH2 Software Family

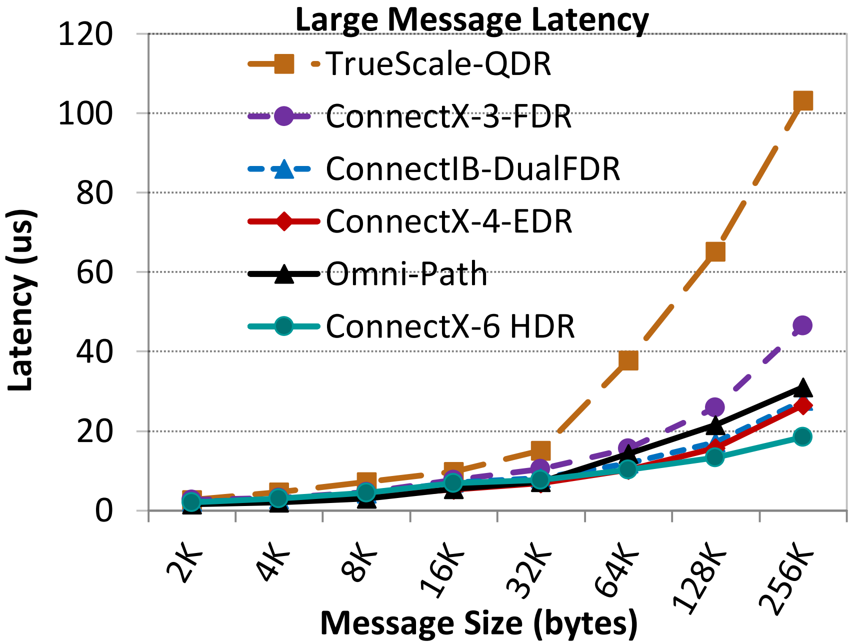
Requirements	Library
MPI with IB, iWARP, Omni-Path, and RoCE	MVAPICH2
Advanced MPI Features/Support, OSU INAM, PGAS and MPI+PGAS with IB, Omni-Path, and RoCE	MVAPICH2-X
MPI with IB, RoCE & GPU and Support for Deep Learning	MVAPICH2-GDR
HPC Cloud with MPI & IB	MVAPICH2-Virt
Energy-aware MPI with IB, iWARP and RoCE	MVAPICH2-EA
MPI Energy Monitoring Tool	OEMT
InfiniBand Network Analysis and Monitoring	OSU INAM
Microbenchmarks for Measuring MPI and PGAS Performance	OMB

Enabling HPC and Deep Learning through MVAPICH2

- High-Performance and Scalable HPC
- CPU-based Deep Learning
- GPU-based Deep Learning

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

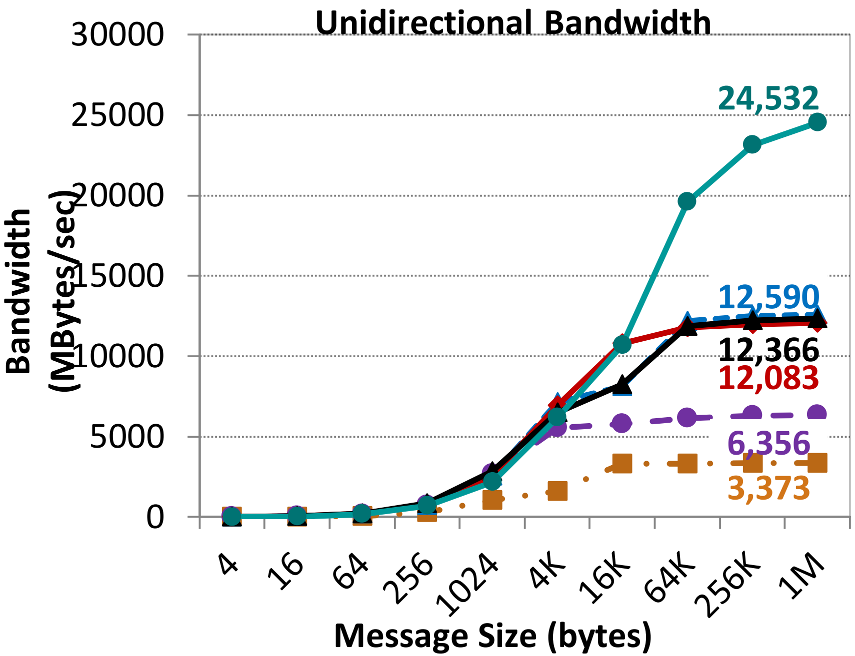
ConnectIB-Dual FDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB switch

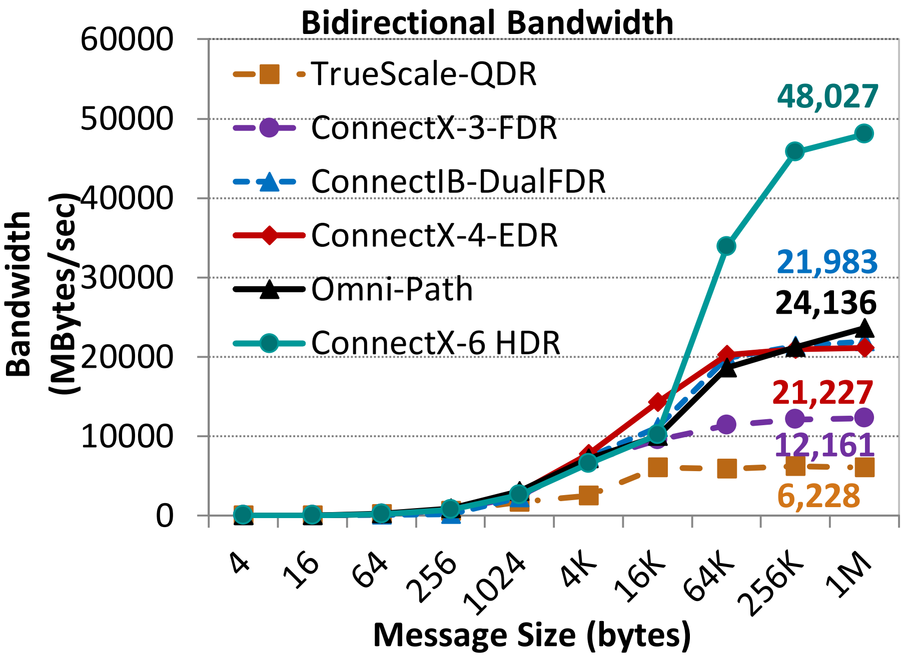
ConnectX-4-EDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB Switch

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

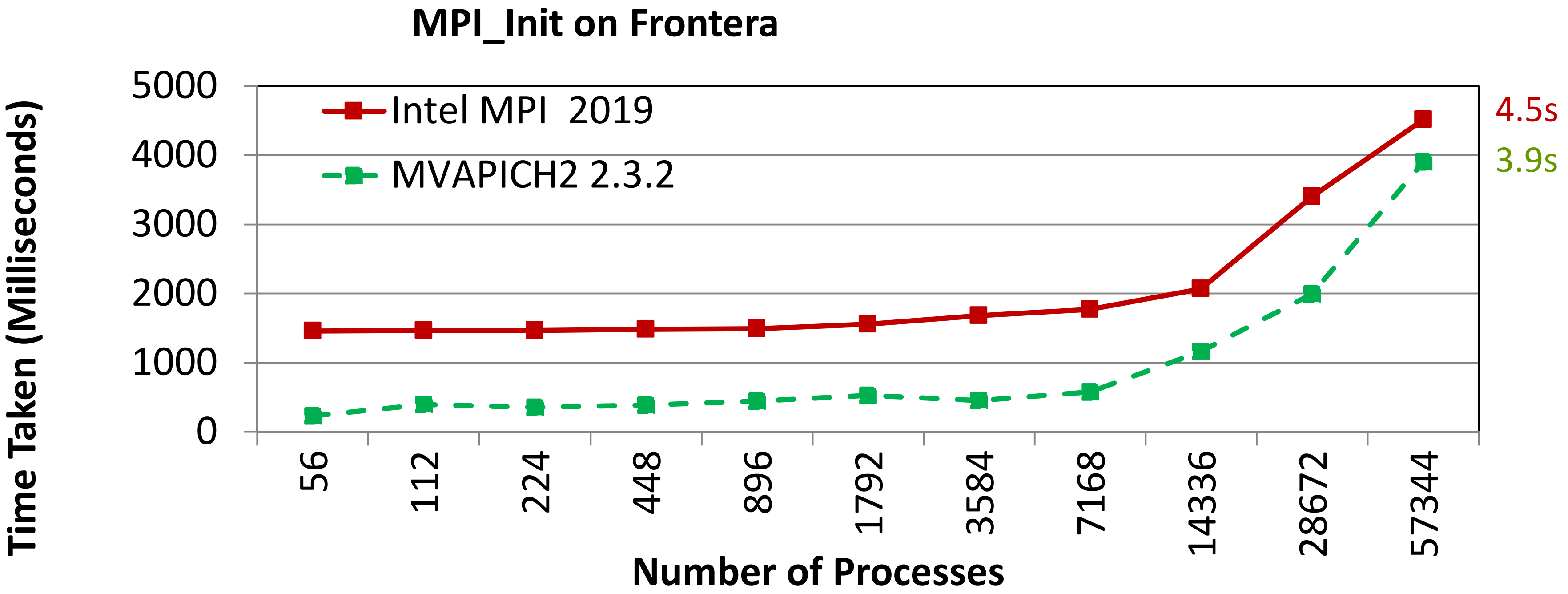
Bandwidth: MPI over IB with MVAPICH2





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

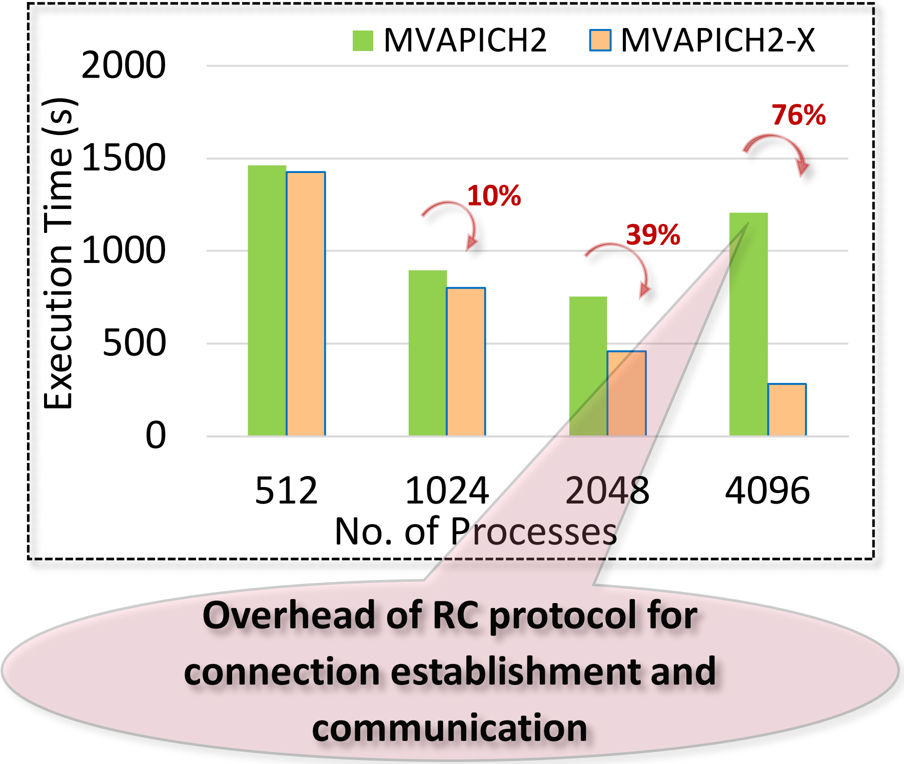


- MPI_Init takes 3.9 seconds on 57,344 processes on 1,024 nodes
- All numbers reported with 56 processes per node

New designs available in MVAPICH2-2.3.2

Impact of Direct Connect (DC) Transport Protocol on Neuron

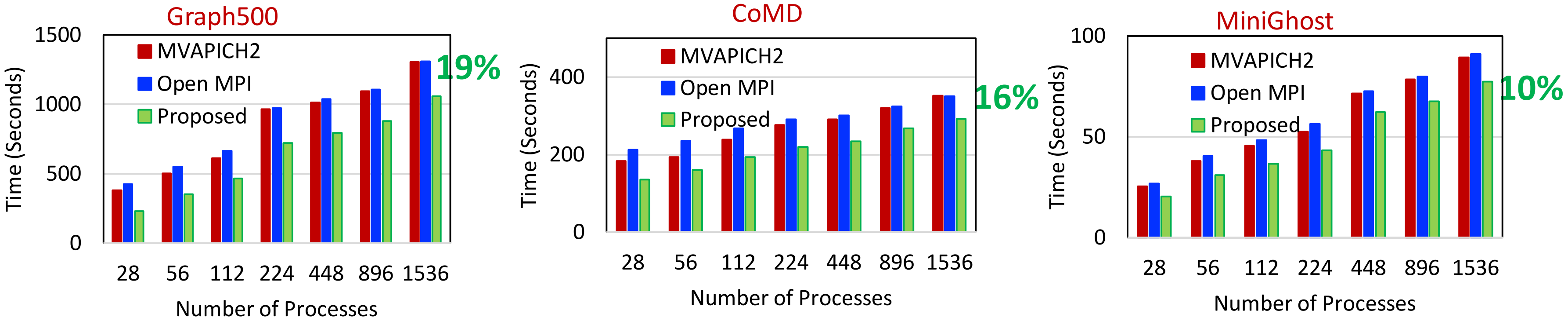
Neuron with YuEtAl2012



- Up to 76% benefits over MVAPICH2 for Neuron using Direct Connected transport protocol at scale
  - VERSION 7.6.2 master (f5a1284) 2018-08-15
- Numbers taken on bbpv2.epfl.ch
  - Knights Landing nodes with 64 ppn
  - ./x86_64/special -mpi -c stop_time=2000 -c is_split=1 parinit.hoc
  - Used "runtime" reported by execution to measure performance
- Environment variables used
  - MV2_USE_DC=1
  - MV2_NUM_DC_TGT=64
  - MV2_SMALL_MSG_DC_POOL=96
  - MV2_LARGE_MSG_DC_POOL=96
  - MV2_USE_RDMA_CM=0

Available from MVAPICH2-X 2.3rc2 onwards

Cooperative Rendezvous Protocols



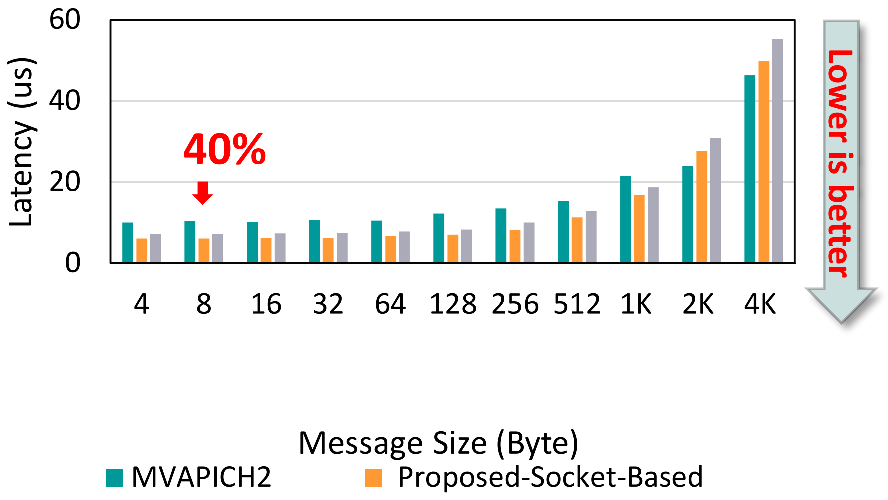
- Use both sender and receiver CPUs to progress communication concurrently
- Dynamically select rendezvous protocol based on communication primitives and sender/receiver availability (load balancing)
- Up to 2x improvement in large message latency and bandwidth
- Up to 19% improvement for Graph500 at 1536 processes

Platform: 2x14 core Broadwell 2680 (2.4 GHz)
Mellanox EDR ConnectX-5 (100 GBps)
Baseline: MVAPICH2X-2.3rc1, Open MPI v3.1.0

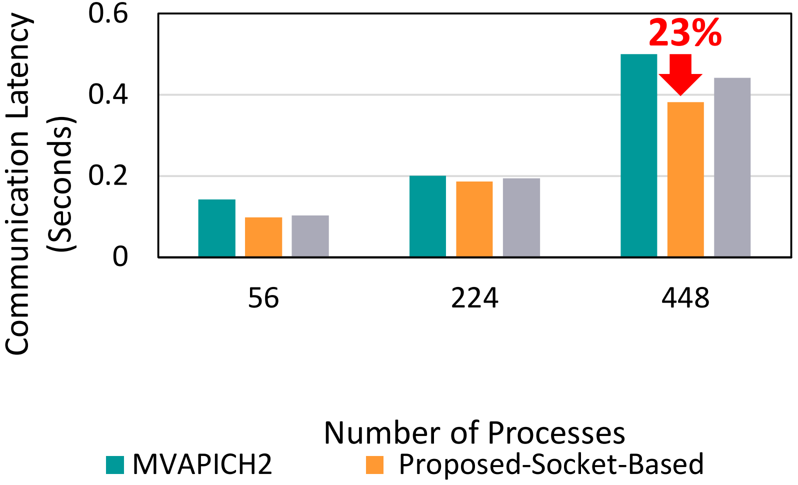
Cooperative Rendezvous Protocols for Improved Performance and Overlap
S. Chakraborty, M. Bayatpour, J Hashmi, H. Subramoni, and DK Panda,
SC '18 (Best Student Paper Award Finalist)

Available in MVAPICH2-X 2.3rc2

Advanced Allreduce Collective Designs Using SHArP and Multi-Leaders



OSU Micro Benchmark (16 Nodes, 28 PPN)

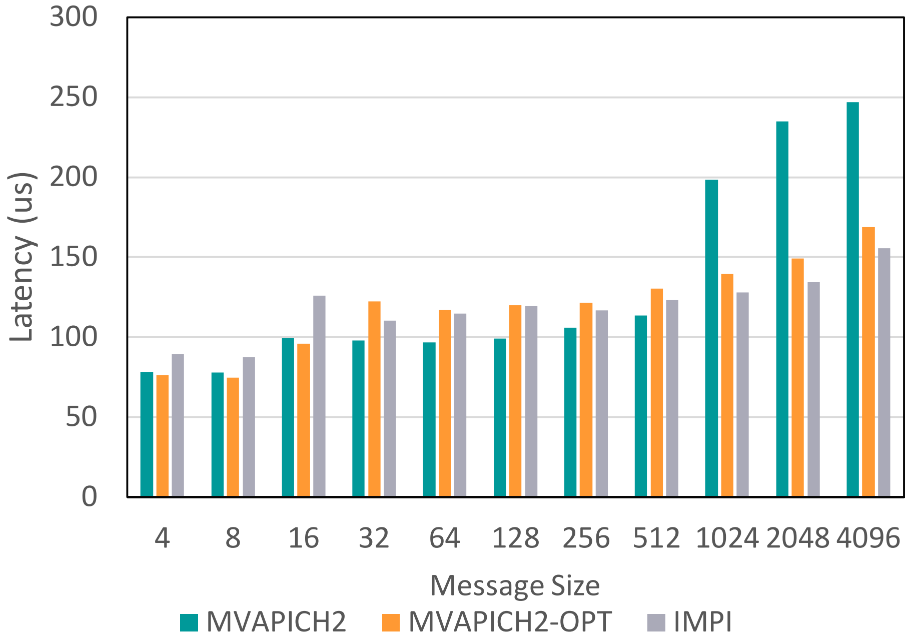


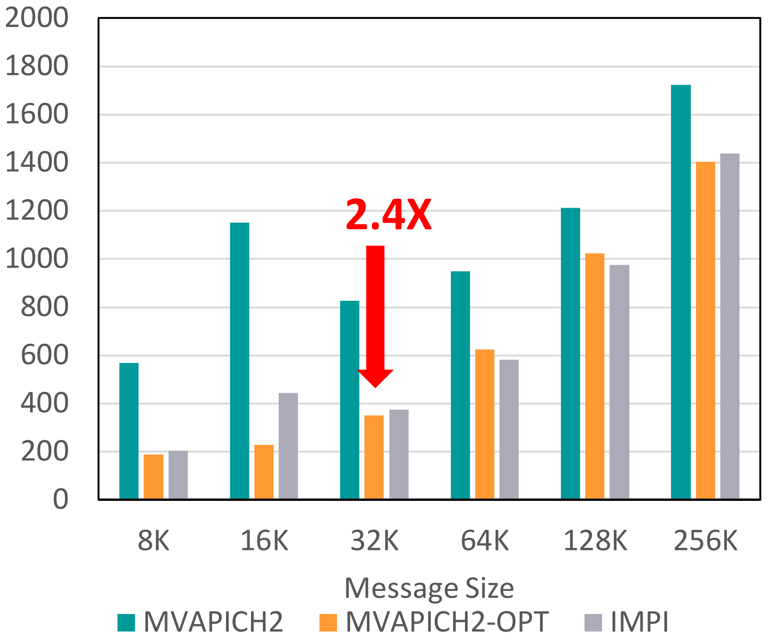
HPCG (28 PPN)

- Socket-based design can reduce the communication latency by 23% and 40% on Broadwell + IB-EDR nodes
- **Support is available since MVAPICH2-X 2.3b**

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

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





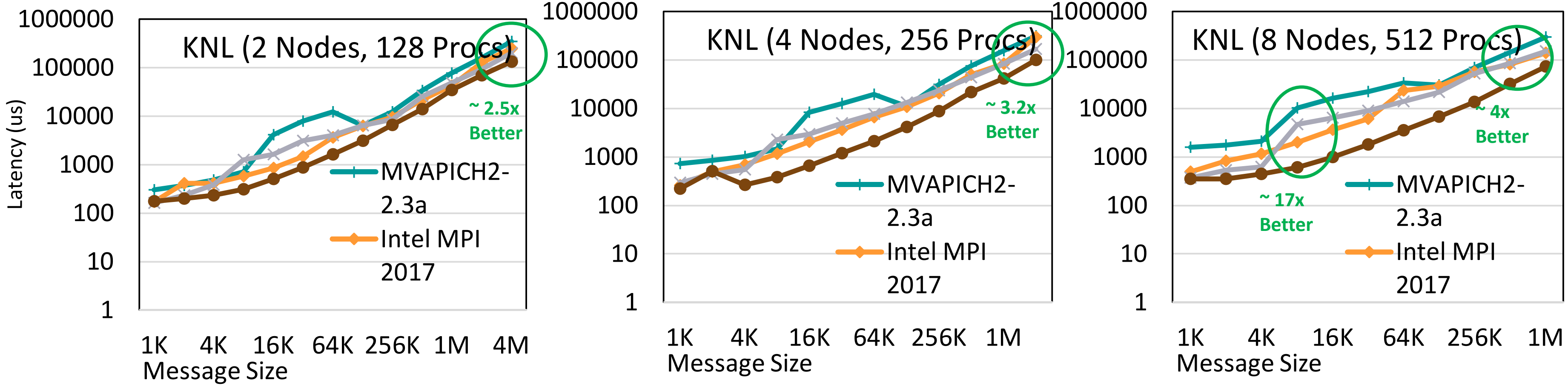
OSU Micro Benchmark 64 PPN

- 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

Optimized CMA-based Collectives for Large Messages



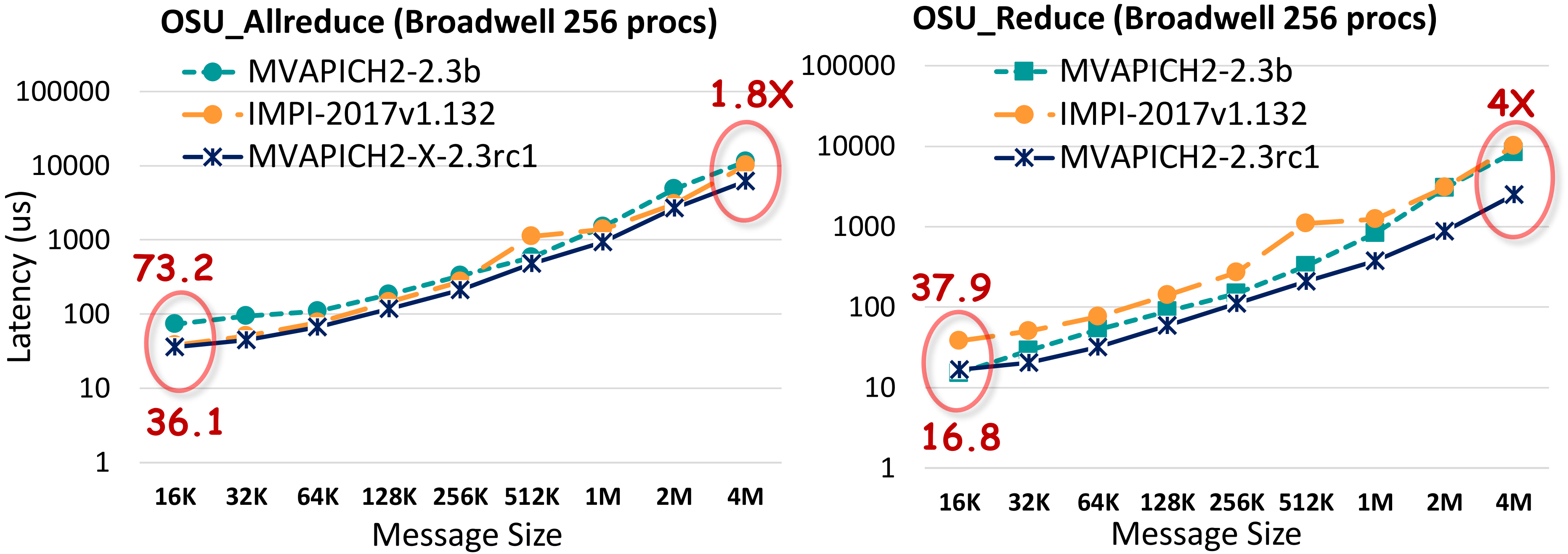
Performance of MPI_Gather on KNL nodes (64PPN)

- Significant improvement over existing implementation for Scatter/Gather with 1MB messages (up to 4x on KNL, 2x on Broadwell, 14x on OpenPOWER)
- New two-level algorithms for better scalability
- Improved performance for other collectives (Bcast, Allgather, and Alltoall)

S. Chakraborty, H. Subramoni, and D. K. Panda, Contention Aware Kernel-Assisted MPI Collectives for Multi/Many-core Systems, *IEEE Cluster '17, BEST Paper Finalist*

Available since MVAPICH2-X 2.3b

Shared Address Space (XPMEM)-based Collectives Design

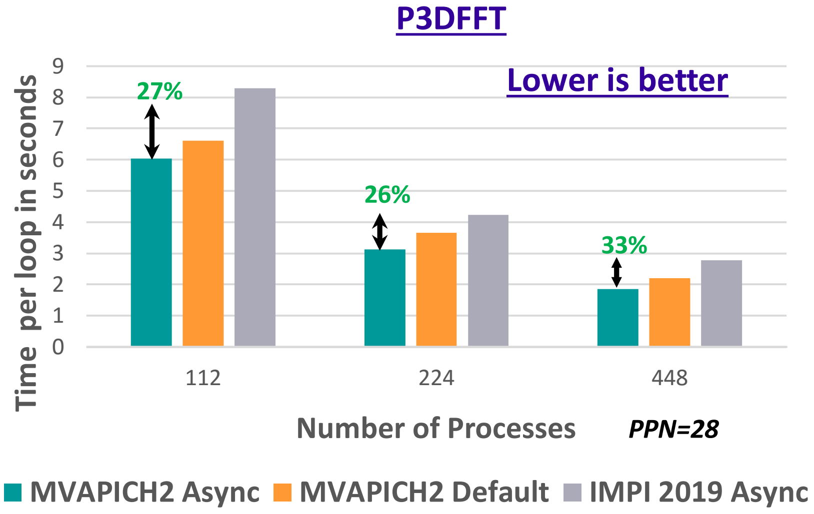


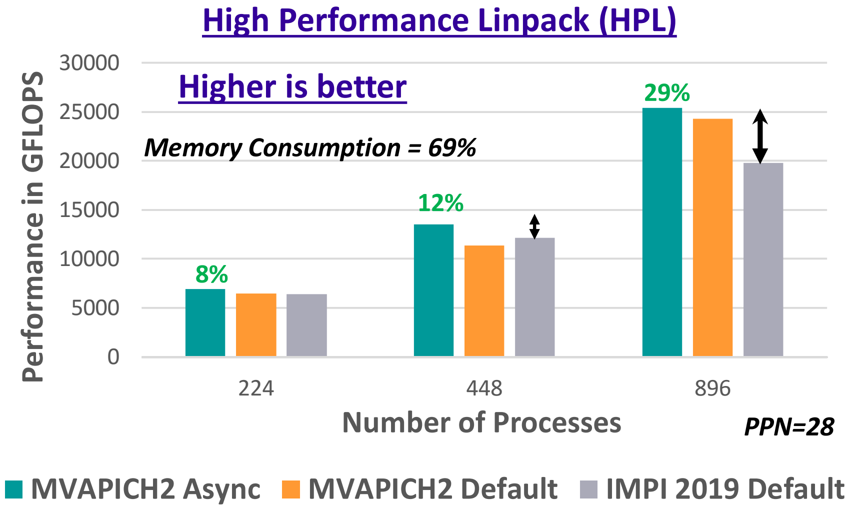
- "*Shared Address Space*"-based true *zero-copy* Reduction collective designs in MVAPICH2
- Offloaded computation/communication to peers ranks in reduction collective operation
- Up to **4X** improvement for 4MB Reduce and up to **1.8X** improvement for 4M AllReduce

J. Hashmi, S. Chakraborty, M. Bayatpour, H. Subramoni, and D. Panda, Designing Efficient Shared Address Space Reduction Collectives for Multi-/Many-cores, International Parallel & Distributed Processing Symposium (IPDPS '18), May 2018.

Available since MVAPICH2-X 2.3rc1

Benefits of Efficient Asynchronous Progress Design: Broadwell + InfiniBand



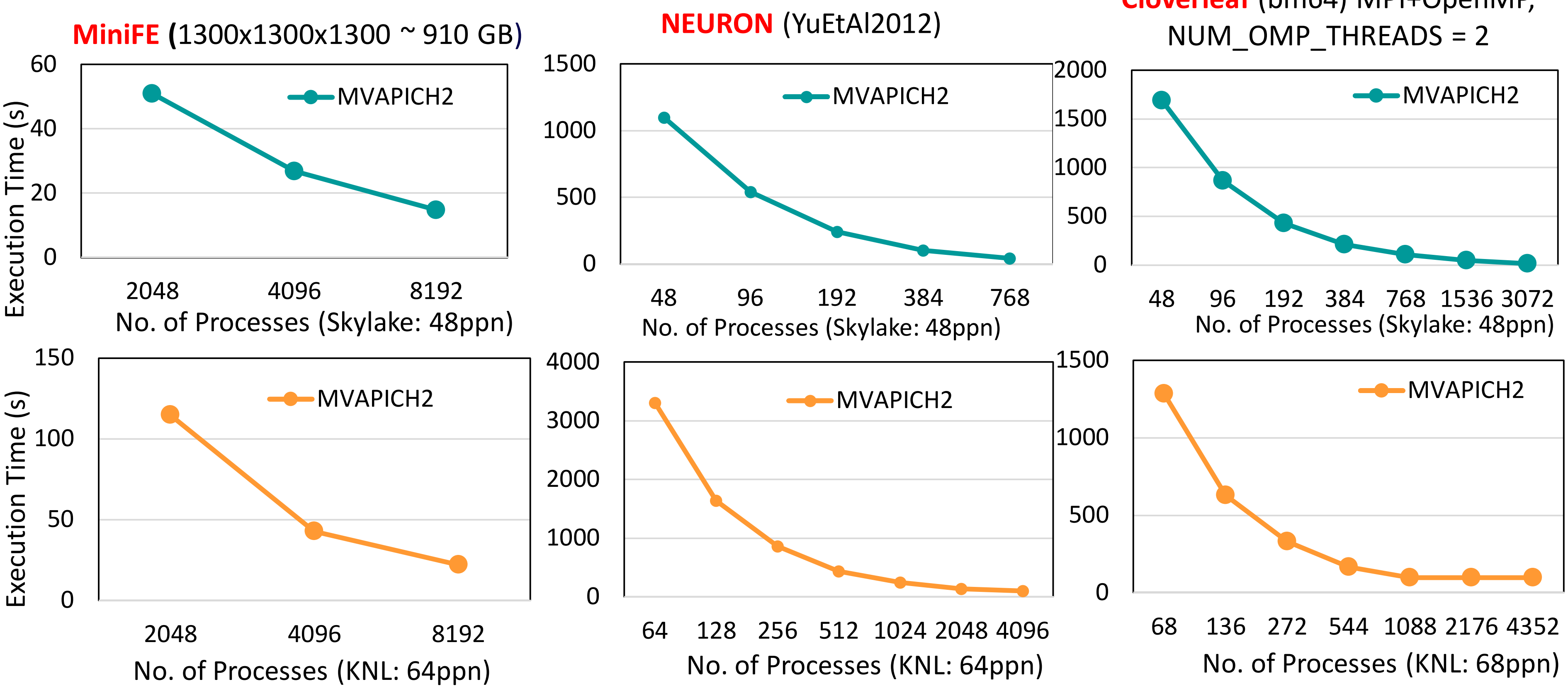


Up to 33% performance improvement in P3DFFT application with 448 processes
Up to 29% performance improvement in HPL application with 896 processes

A. Ruhela, H. Subramoni, S. Chakraborty, M. Bayatpour, P. Kousha, and D.K. Panda, Efficient Asynchronous Communication Progress for MPI without Dedicated Resources, EuroMPI 2018. Enhanced version accepted for PARCO Journal.

Available since MVAPICH2-X 2.3rc1

Application Scalability on Skylake and KNL (Stampede2)



Courtesy: Mahidhar Tatineni @SDSC, Dong Ju (DJ) Choi@SDSC, and Samuel Khuvis@OSC ---- Testbed: TACC Stampede2 using MVAPICH2-2.3b

Runtime parameters: *MV2_SMPI_LENGTH_QUEUE=524288 PSM2_MQ_RNDV_SHM_THRESH=128K PSM2_MQ_RNDV_HFI_THRESH=128K*

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

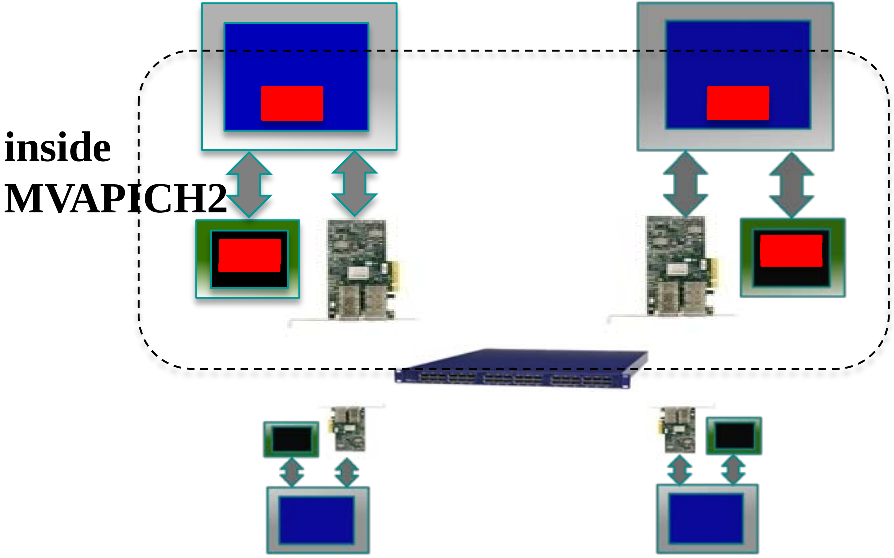
At Sender:

MPI_Send(s_devbuf, size, …);

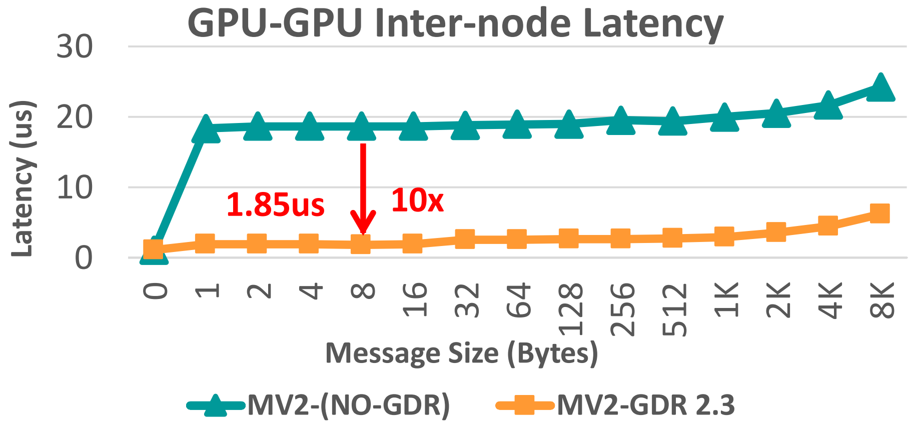
At Receiver:

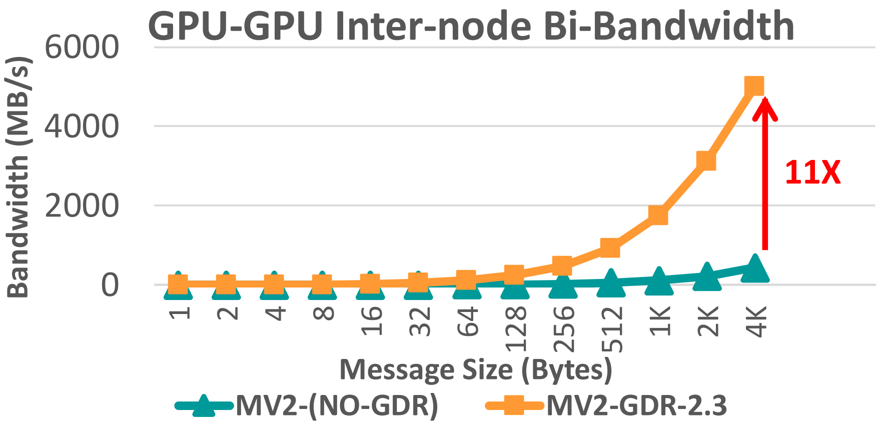
MPI_Recv(r_devbuf, size, …);

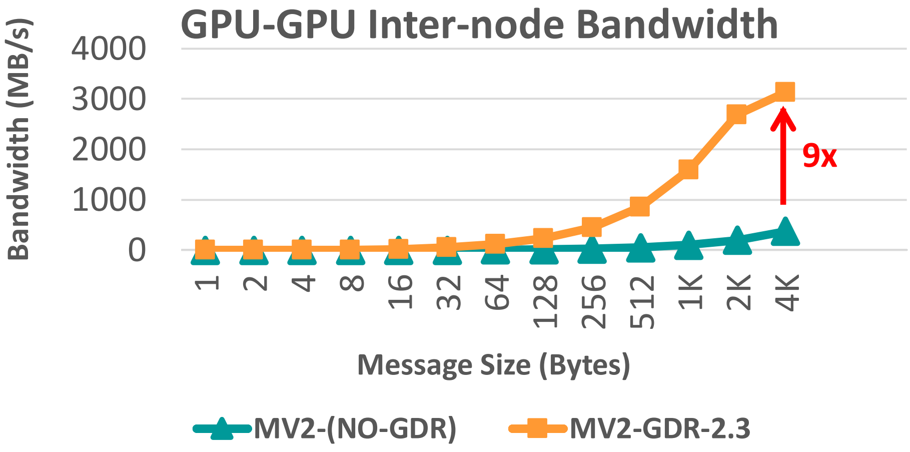
High Performance and High Productivity



Optimized MVAPICH2-GDR Design

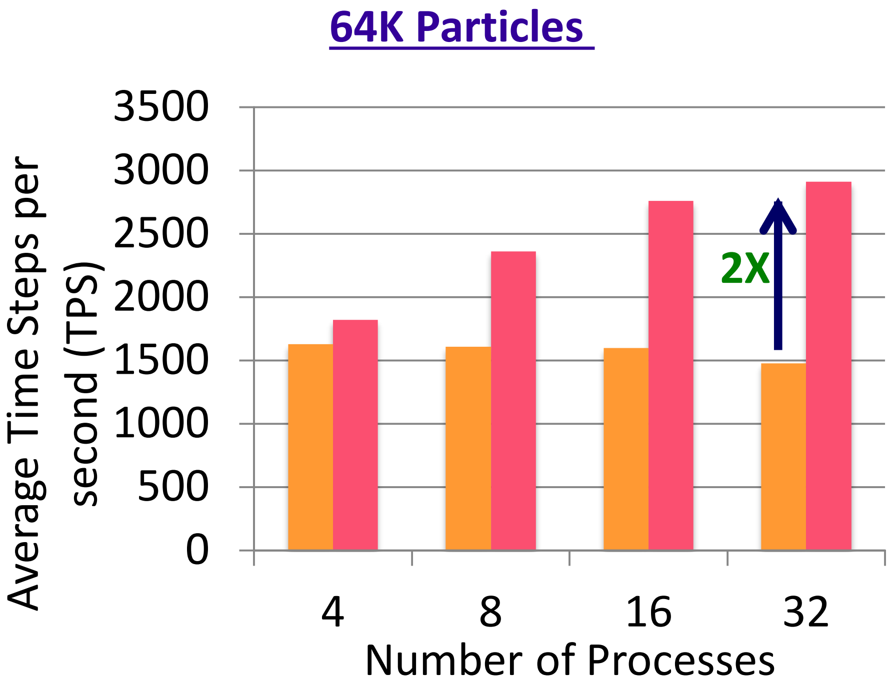


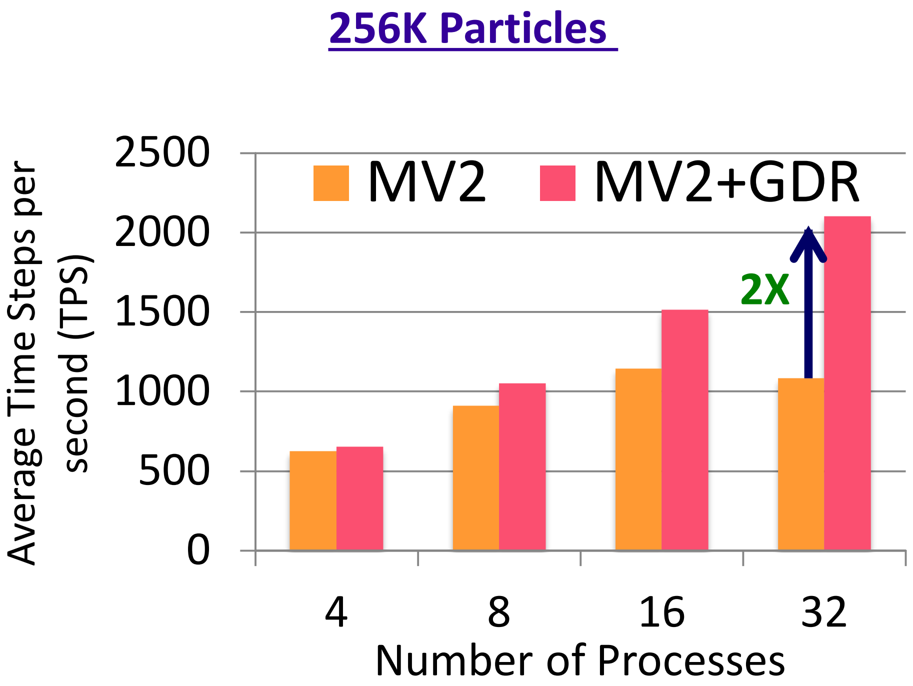




MVAPICH2-GDR-2.3
Intel Haswell (E5-2687W @ 3.10 GHz) node - 20 cores
NVIDIA Volta V100 GPU
Mellanox Connect-X4 EDR HCA
CUDA 9.0
Mellanox OFED 4.0 with GPU-Direct-RDMA

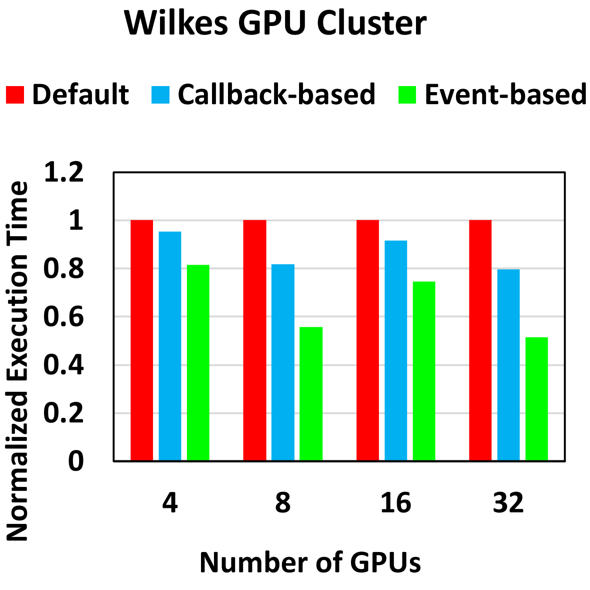
Application-Level Evaluation (HOOMD-blue)

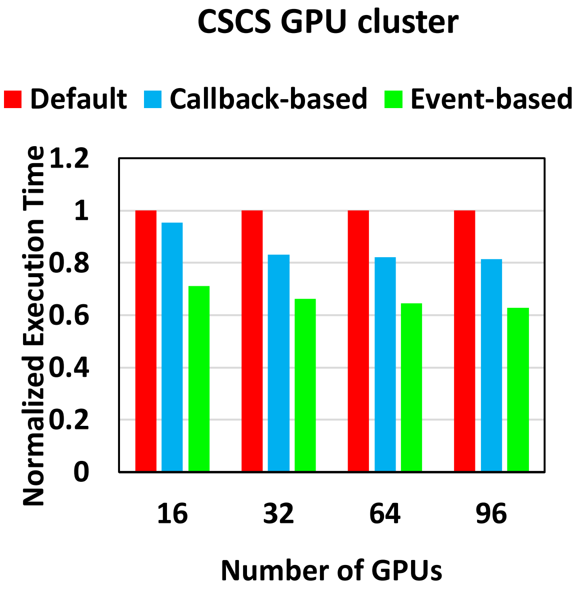


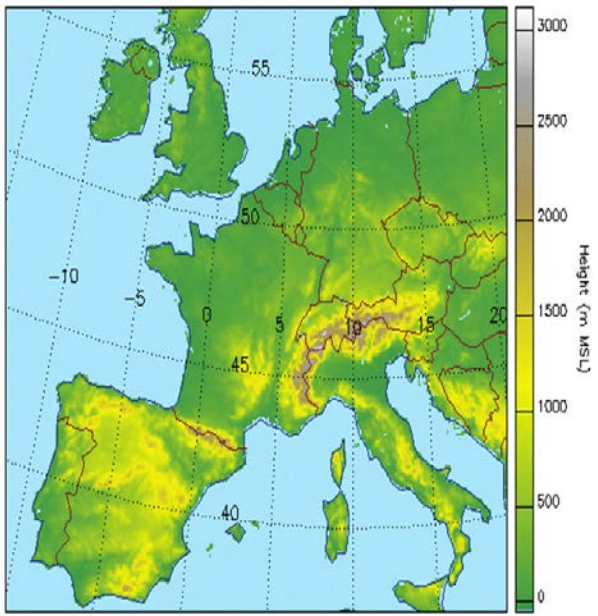


- Platform: Wilkes (Intel Ivy Bridge + NVIDIA Tesla K20c + Mellanox Connect-IB)
- HoomdBlue Version 1.0.5
  - GDRCOPY enabled: MV2_USE_CUDA=1 MV2_IBA_HCA=mlx5_0 MV2_IBA_EAGER_THRESHOLD=32768 MV2_VBUF_TOTAL_SIZE=32768 MV2_USE_GPUDIRECT_LOOPBACK_LIMIT=32768 MV2_USE_GPUDIRECT_GDRCOPY=1 MV2_USE_GPUDIRECT_GDRCOPY_LIMIT=16384

Application-Level Evaluation (Cosmo) and Weather Forecasting in Switzerland







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

Cosmo model: http://www2.cosmo-model.org/content/tasks/operational/meteoSwiss/

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

C. Chu, K. Hamidouche, A. Venkatesh, D. Banerjee , H. Subramoni, and D. K. Panda, Exploiting Maximal Overlap for Non-Contiguous Data Movement Processing on Modern GPU-enabled Systems, IPDPS'16

MVAPICH2-Azure 2.3.2

- **Released on 08/16/2019**
- Major Features and Enhancements
  - **Based on MVAPICH2-2.3.2**
  - **Enhanced tuning for point-to-point and collective operations**
  - **Targeted for Azure HB & HC virtual machine instances**
  - **Flexibility for 'one-click' deployment**
  - **Tested with Azure HB & HC VM instances**

MVAPICH2-X-AWS 2.3

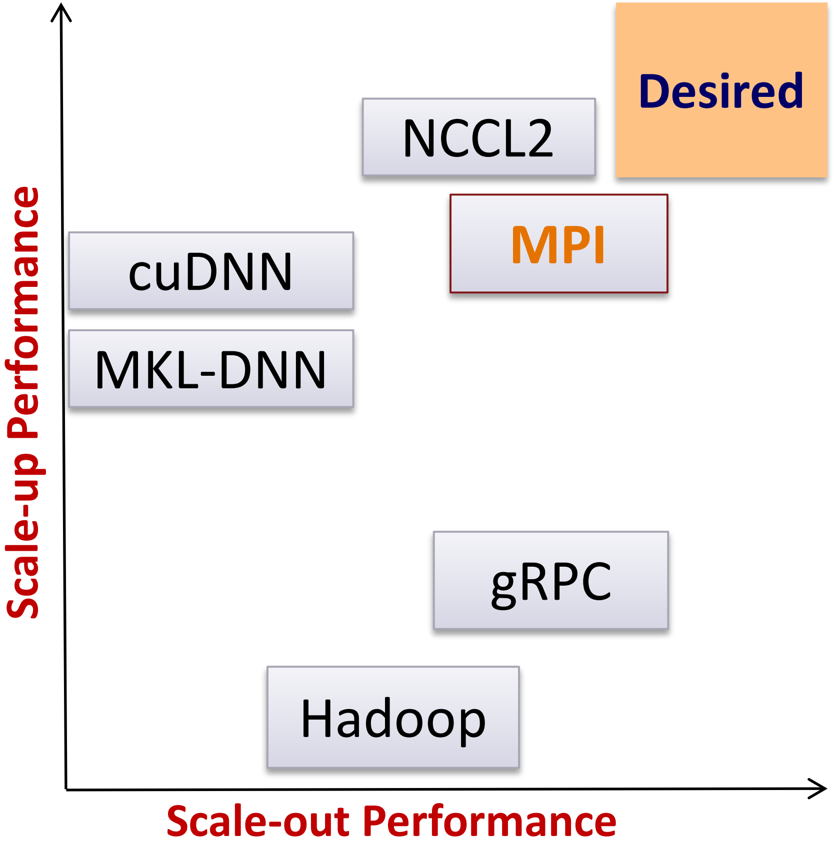
- **Released on 08/12/2019**
- Major Features and Enhancements
  - **Based on MVAPICH2-X 2.3**
  - **New design based on Amazon EFA adapter's Scalable Reliable Datagram (SRD) transport protocol**
  - **Support for XPMEM based intra-node communication for point-to-point and collectives**
  - **Enhanced tuning for point-to-point and collective operations**
  - **Targeted for AWS instances with Amazon Linux 2 AMI and EFA support**
  - **Tested with c5n.18xlarge instance**

Enabling HPC and Deep Learning through MVAPICH2

- High-Performance and Scalable HPC
- CPU-based Deep Learning
- GPU-based Deep Learning

Deep Learning: New Challenges for MPI Runtimes

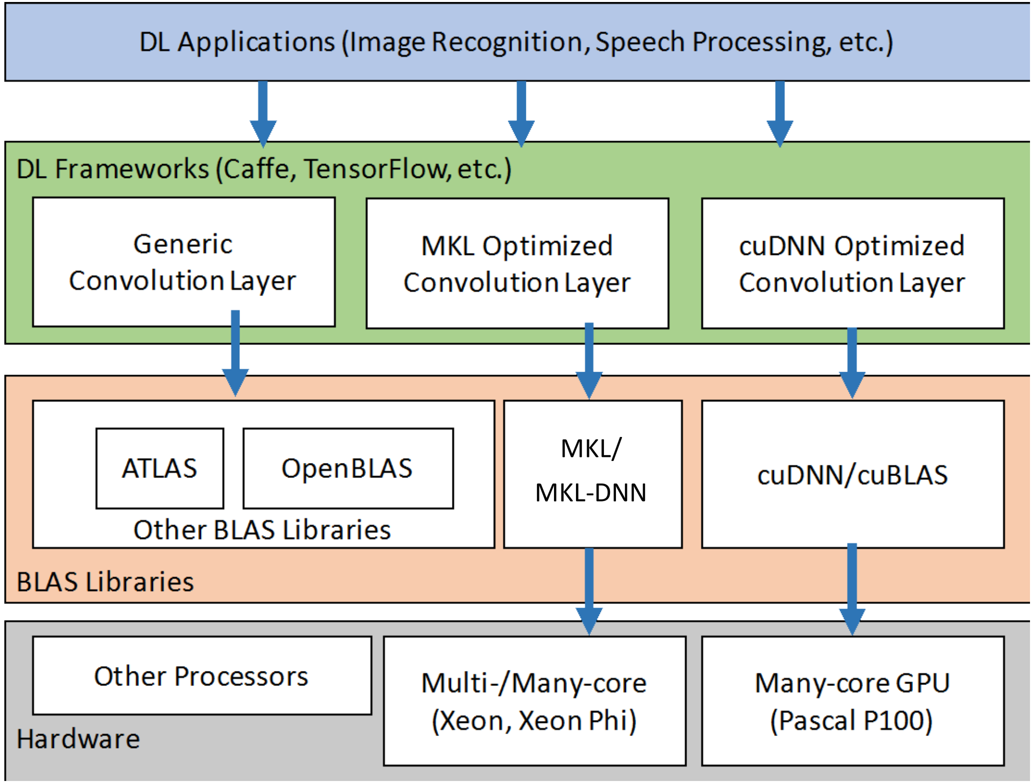
- Deep Learning frameworks are a different game altogether
  - Unusually large message sizes (order of megabytes)
  - Most communication based on GPU buffers
- Existing State-of-the-art
  - cuDNN, cuBLAS, NCCL --> **scale-up** performance
  - NCCL2, CUDA-Aware MPI --> **scale-out** performance
    - For small and medium message sizes only!
- Can we **optimize** the MPI runtime (**MVAPICH2-X and MVAPICH2-GDR**) for DL frameworks?
  - Efficient **Overlap** of Computation and Communication
  - Efficient **Large-Message** Communication (Reductions)
- What **application co-designs** are needed to exploit **communication-runtime co-designs**?



A. A. Awan, K. Hamidouche, J. M. Hashmi, and D. K. Panda, S-Caffe: Co-designing MPI Runtimes and Caffe for Scalable Deep Learning on Modern GPU Clusters. In *Proceedings of the 22nd ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming* (PPoPP '17)

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



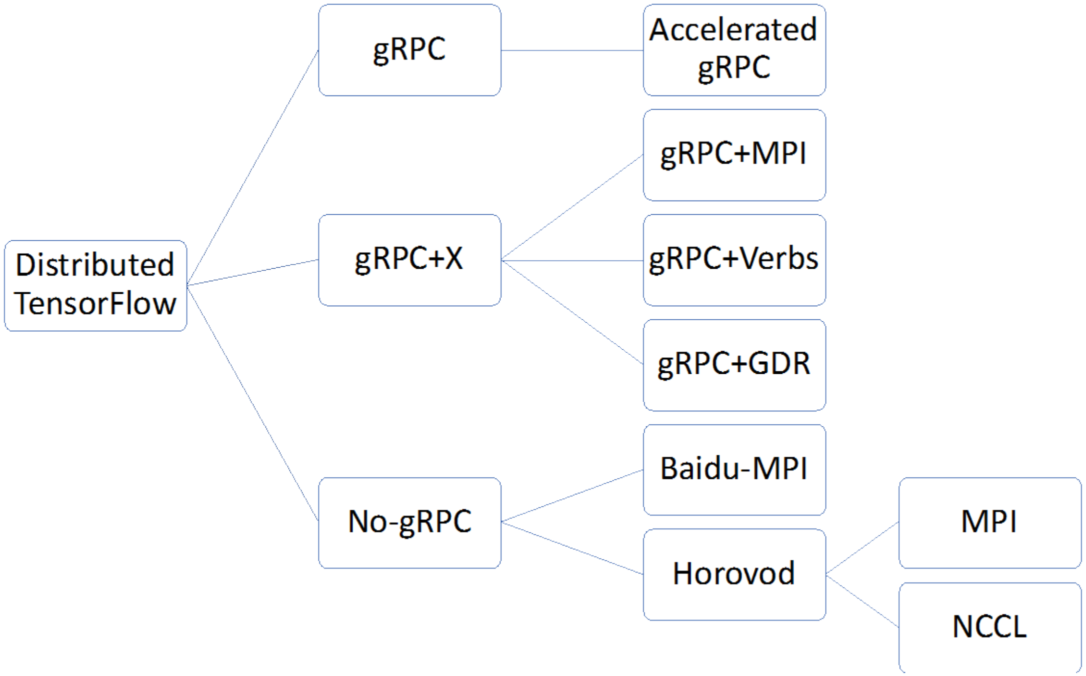
A. A. Awan, H. Subramoni, and Dhabaleswar K. Panda. "An In-depth Performance Characterization of CPU- and GPU-based DNN Training on Modern Architectures", In Proceedings of the Machine Learning on HPC Environments (MLHPC'17). ACM, New York, NY, USA, Article 8.

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)

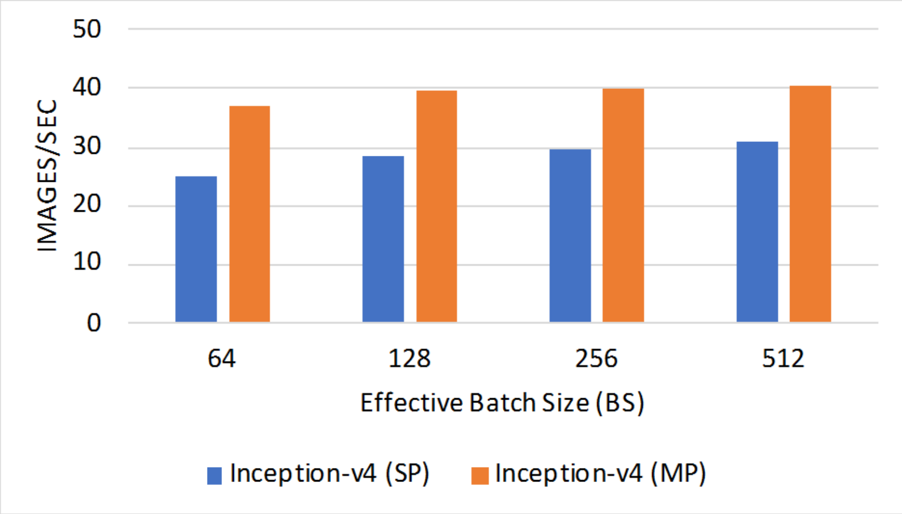
- TensorFlow is the most popular DL framework
- gRPC is the official distributed training runtime
  - Many problems for HPC use-cases
- Community efforts - Baidu and Uber’s Horovod have added MPI support to TF across nodes
- Need to understand several options currently available →

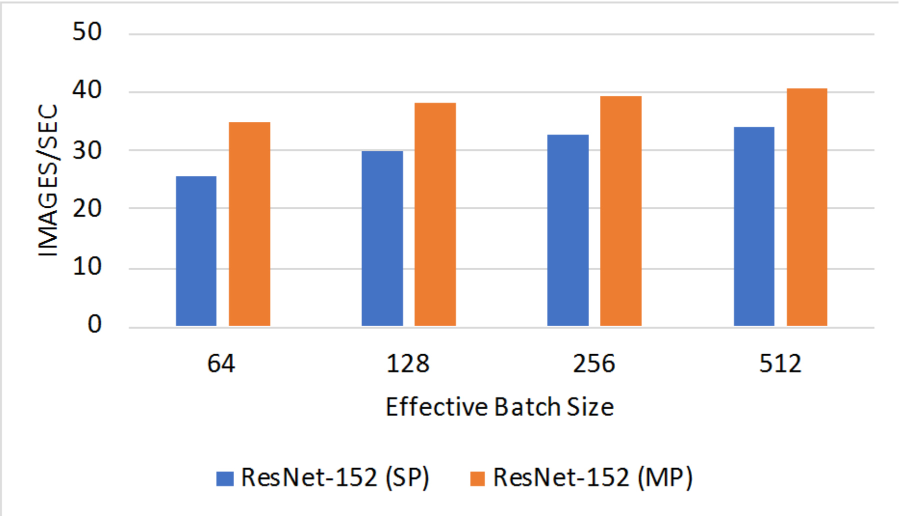


A. Awan, J. Bedorf, C. Chu, H. Subramoni and D. K. Panda, “Scalable Distributed DNN Training using TensorFlow and CUDA-Aware MPI: Characterization, Designs, and Performance Evaluation, CCGrid ‘19. https://arxiv.org/abs/1810.11112

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

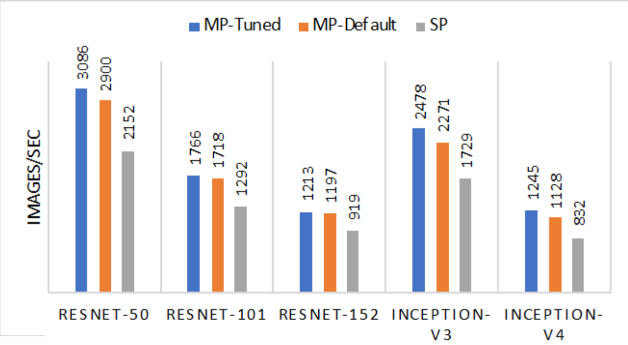


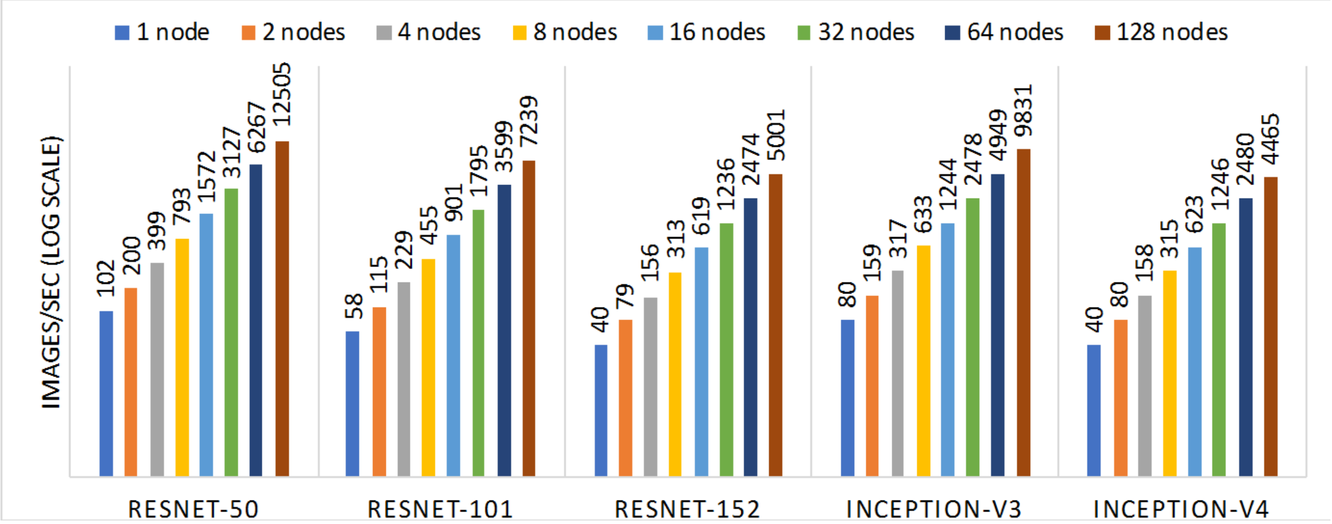


A. Jain, A. Awan, Q. Anthony, H. Subramoni, and D. K. Panda, Performance Characterization of DNN Training using TensorFlow and PyTorch on Modern Clusters, Cluster '19.

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





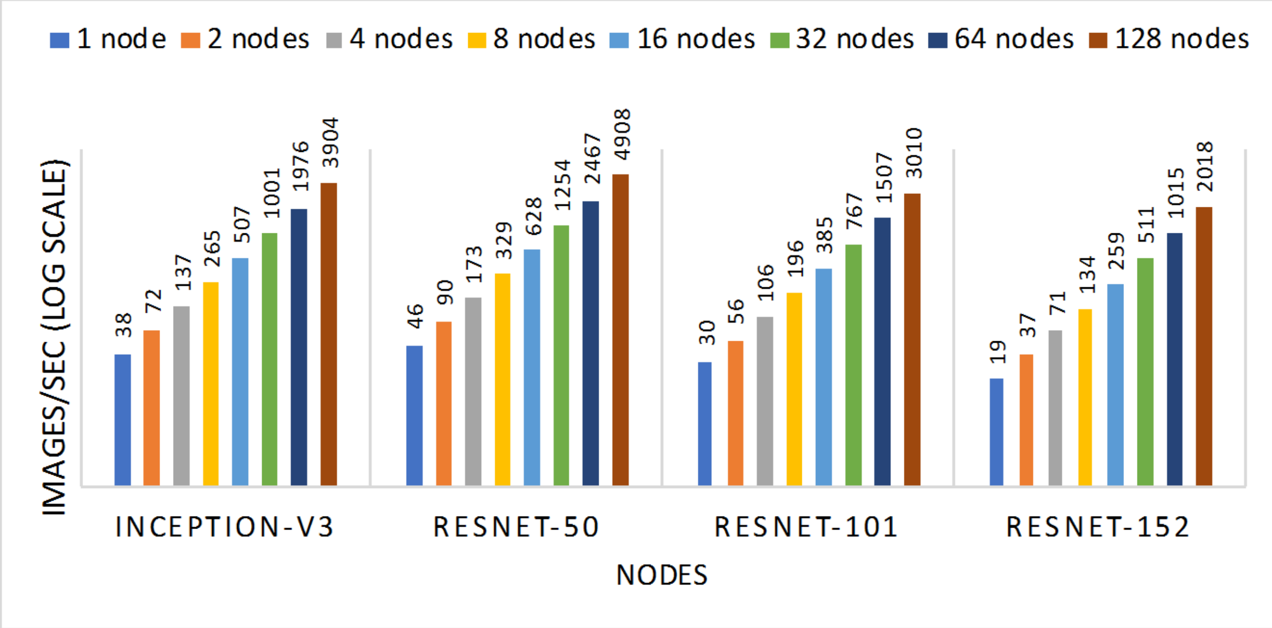
←

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

A. Jain, A. Awan, Q. Anthony, H. Subramoni, and D. K. Panda, Performance Characterization of DNN Training using TensorFlow and PyTorch on Modern Clusters, Cluster '19.

Multi-node Performance for PyTorch

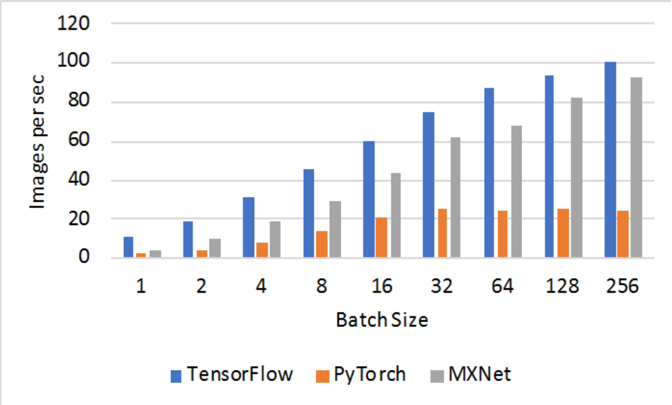
- Early results with PyTorch (using tuned configuration)
  - Good scaling (106X speedup on 128 nodes)
  - Overall -- Slower than TensorFlow



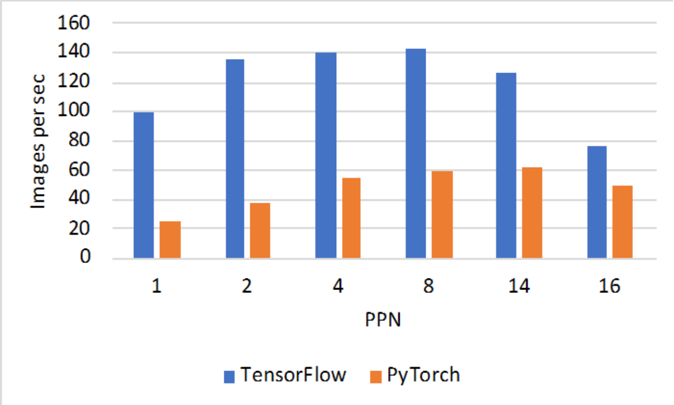
A. Jain, A. Awan, Q. Anthony, H. Subramoni, and D. K. Panda, Performance Characterization of DNN Training using TensorFlow and PyTorch on Modern Clusters, Cluster '19.

Deep Learning on TACC Frontera

- TensorFlow, PyTorch, and MXNet are widely used Deep Learning Frameworks
- Optimized by Intel using Math Kernel Library for DNN (MKL-DNN) for Intel processors
- Single Node performance can be improved by running Multiple MPI processes



Impact of Batch Size on Performance for ResNet-50

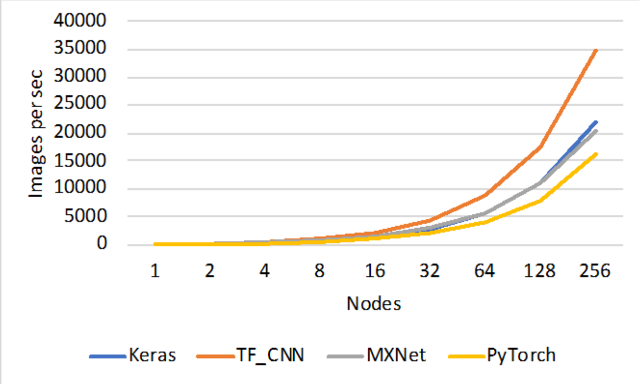


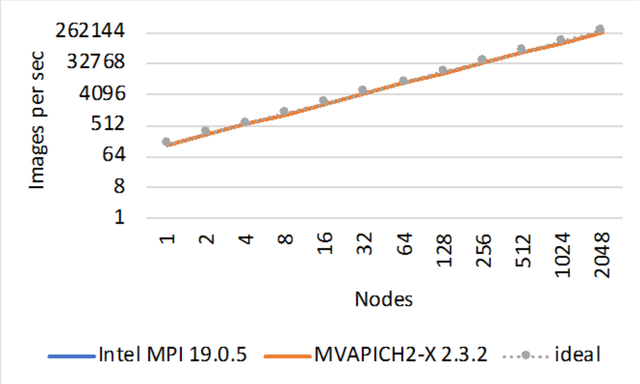
Performance Improvement using Multiple MPI processes

A. Jain et al., Scaling Deep Learning Frameworks on Frontera using MVAPICH2 MPI, under review

Deep Learning on TACC Frontera

- Observed 260K images per sec for ResNet-50 on 2,048 Nodes
- Scaled MVAPICH2-X on 2,048 nodes on Frontera for Distributed Training using TensorFlow
- ResNet-50 can be trained in 7 minutes on 2048 nodes (114,688 cores)





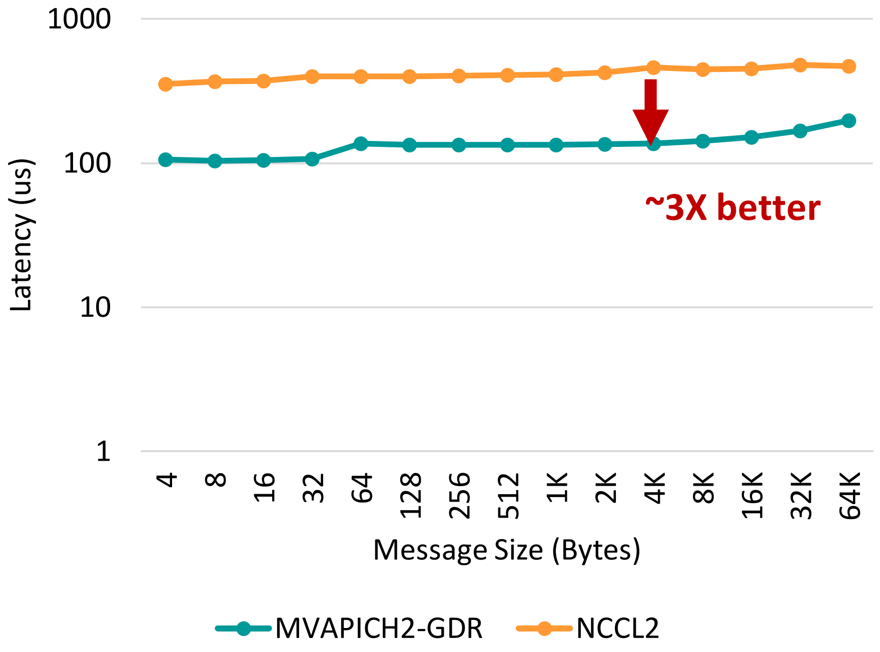
A. Jain et al., Scaling Deep Learning Frameworks on Frontera using MVAPICH2 MPI, under review

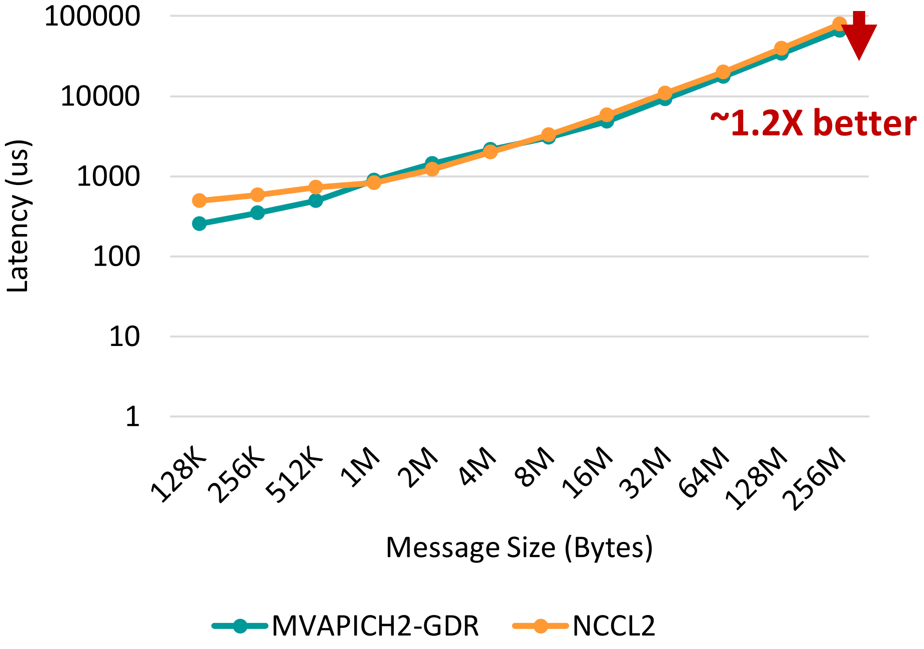
Enabling HPC and Deep Learning through MVAPICH2

- High-Performance and Scalable HPC
- CPU-based Deep Learning
- GPU-based Deep Learning

MVAPICH2-GDR vs. NCCL2 – Allreduce Operation

- **Optimized designs in MVAPICH2-GDR 2.3 offer better/comparable performance for most cases**
- **MPI_Allreduce (MVAPICH2-GDR) vs. ncclAllreduce (NCCL2) on 16 GPUs**

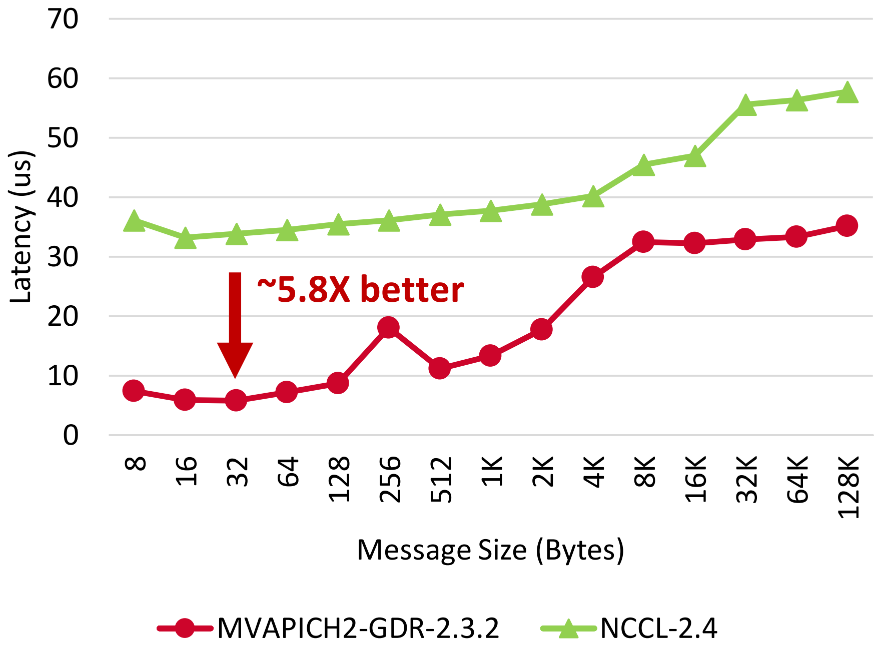


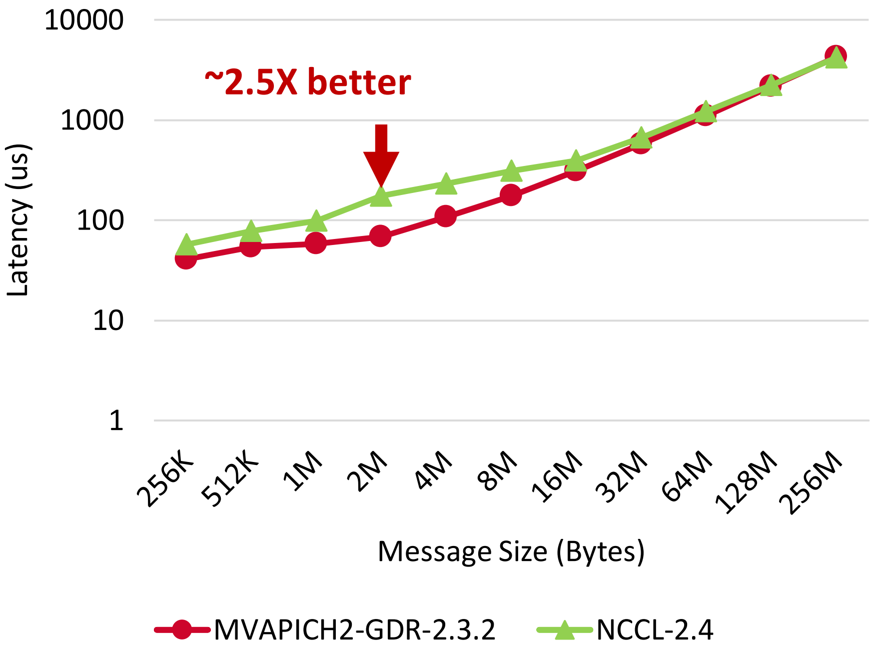


Platform: Intel Xeon (Broadwell) nodes equipped with a dual-socket CPU, 1 K-80 GPUs, and EDR InfiniBand Inter-connect

MVAPICH2-GDR vs. NCCL2 – Allreduce Operation (DGX-2)

- **Optimized designs in MVAPICH2-GDR offer better/comparable performance for most cases**
- **MPI_Allreduce (MVAPICH2-GDR) vs. ncclAllreduce (NCCL2) on 1 DGX-2 node (16 Volta GPUs)**

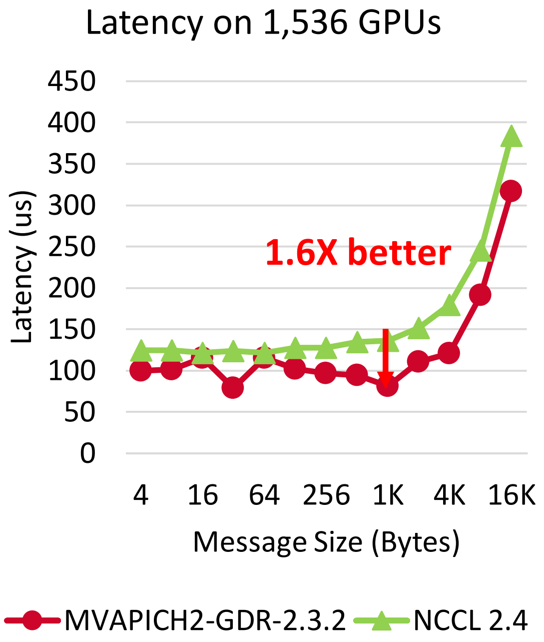


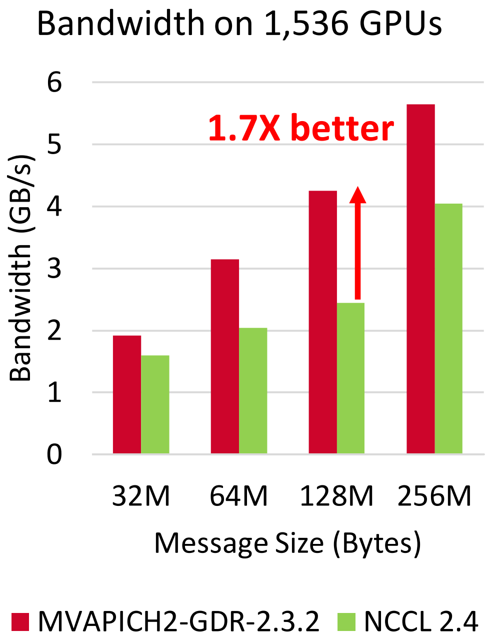


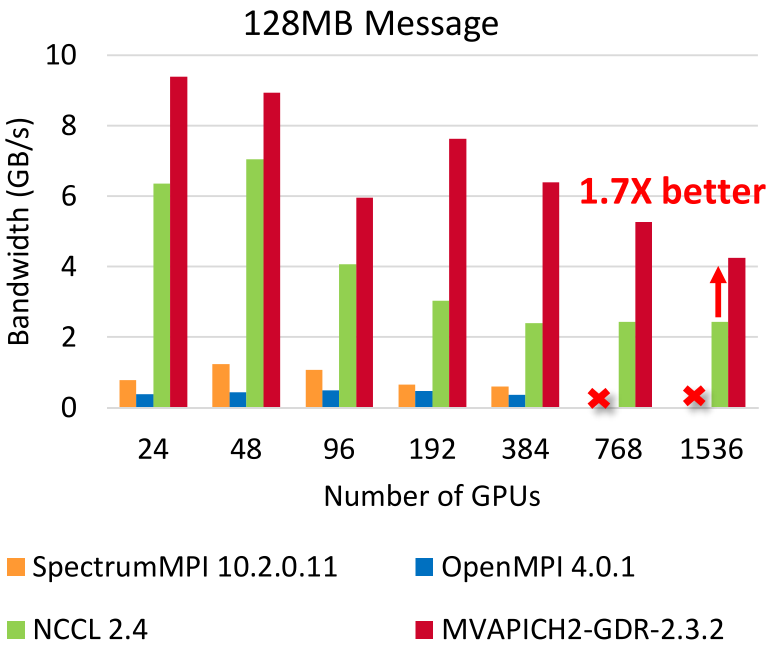
Platform: Nvidia DGX-2 system (16 Nvidia Volta GPUs connected with NVSwitch), CUDA 9.2

MVAPICH2-GDR: Enhanced MPI_Allreduce at Scale

- Optimized designs in MVAPICH2-GDR offer better performance for most cases
- MPI_Allreduce (MVAPICH2-GDR) vs. ncclAllreduce (NCCL2) up to 1,536 GPUs



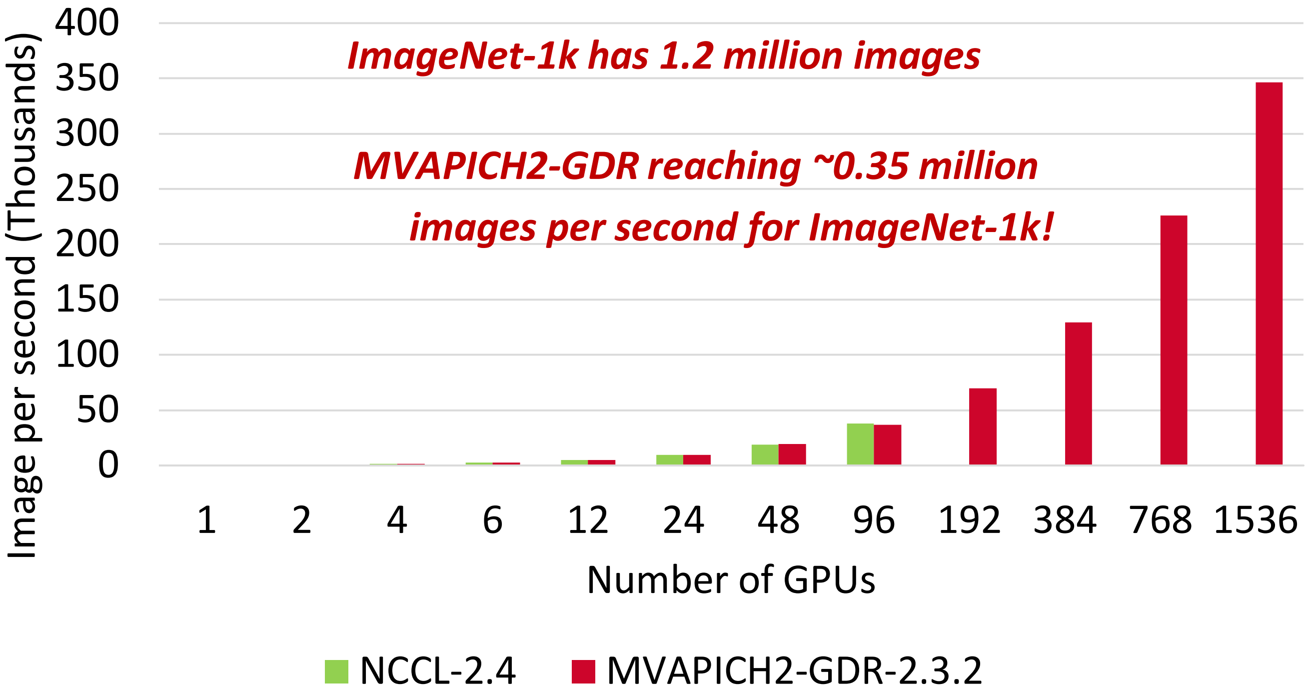




Platform: Dual-socket IBM POWER9 CPU, 6 NVIDIA Volta V100 GPUs, and 2-port InfiniBand EDR Interconnect

Distributed Training with TensorFlow and MVAPICH2-GDR

- ResNet-50 Training using TensorFlow benchmark on SUMMIT -- 1536 Volta GPUs!
- 1,281,167 (1.2 mil.) images
- Time/epoch = 3.6 seconds
- Total Time (90 epochs) = 3.6 x 90 = 332 seconds = **5.5 minutes!**



*We observed errors for NCCL2 beyond 96 GPUs

Platform: The Summit Supercomputer (#1 on Top500.org) – 6 NVIDIA Volta GPUs per node connected with NVLink, CUDA 9.2

Conclusions

- Support for Scalable HPC and Deep Learning is getting important
- Requires high-performance middleware designs while exploiting modern interconnects and multi-core processors
- Provided an overview of MVAPICH2 MPI library to achieve scalable HPC and Deep Learning
- Will continue to enable the HPC and DL community to achieve scalability and high-performance for their workloads

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 provided to Lawrence Livermore National Laboratory (LLNL) for the last two years

X-ScaleSolutions

Silver ISV Member for the OpenPOWER Consortium + Products

- Has joined the OpenPOWER Consortium as a silver ISV member
- 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 OpenPOWER North America Summit)
  - X-ScaleHPC
  - X-ScaleAI
  - Send an e-mail to contactus@x-scalesolutions.com for free trial!!

X-ScaleSolutions

7th Annual MVAPICH User Group (MUG) Meeting

- **August 19-21, 2019; Columbus, Ohio, USA**
- **Keynote Speakers**
  - Dan Stanzione, Texas Advanced Computing Center (TACC)
  - Robert Harrison, Director of the Institute of Advanced Computational Science (IACS) and Brookhaven Computational Science Center (CSC)
- **Tutorials**
  - ARM
  - IBM
  - Mellanox
  - OSU/MVAPICH2
- **Invited Speakers**
  - Gregory Blum Becker, Lawrence Livermore National Laboratory
  - Nicholas Brown, EPCC, The University of Edinburg (United Kingdom)
  - Gene Cooperman, Northeastern University
  - Hyon-Wook Jin, Konkuk University (South Korea)
  - Jithin Jose, Microsoft Azure
  - Minsik Kim, KISTI Supercomputing Center (South Korea)
  - Pramod Kumbhar, Blue Brain Project, EPFL (Switzerland)
  - Naoya Maruyama, Lawrence Livermore National Laboratory
  - Heechang Na, Ohio Supercomputer Center
  - Vikram Saletore, Intel
  - Jeffrey Salmond, University of Cambridge (United Kingdom)
  - Gilad Shainer, Mellanox
  - Sameer Shende, Paratools and University of Oregon
  - Sayantan Sur, Intel
  - Shinichiro Takizawa, RWBC-OIL, AIST (Japan)
  - Mahidhar Tatineni, San Diego Supercomputing Center (SDSC)
  - Karen Tomko, Ohio Supercomputer Center

Slides and Videos of the talks are available from

http://mug.mvapich.cse.ohio-state.edu

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 Students (Undergraduate)

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

Current Post-doc

- 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.)
- 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

Multiple Positions Available in My Group

- Looking for Bright and Enthusiastic Personnel to join as
  - PhD Students
  - Post-Doctoral Researchers
  - MPI Programmer/Software Engineer
  - Hadoop/Big Data Programmer/Software Engineer
  - Deep Learning and Cloud Programmer/Software Engineer
- If interested, please send an e-mail to panda@cse.ohio-state.edu

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/