# Designing Scalable HPC, Deep Learning, Big Data, and Cloud Middleware for Exascale Systems

Talk at SCEC ’18 Workshop

by

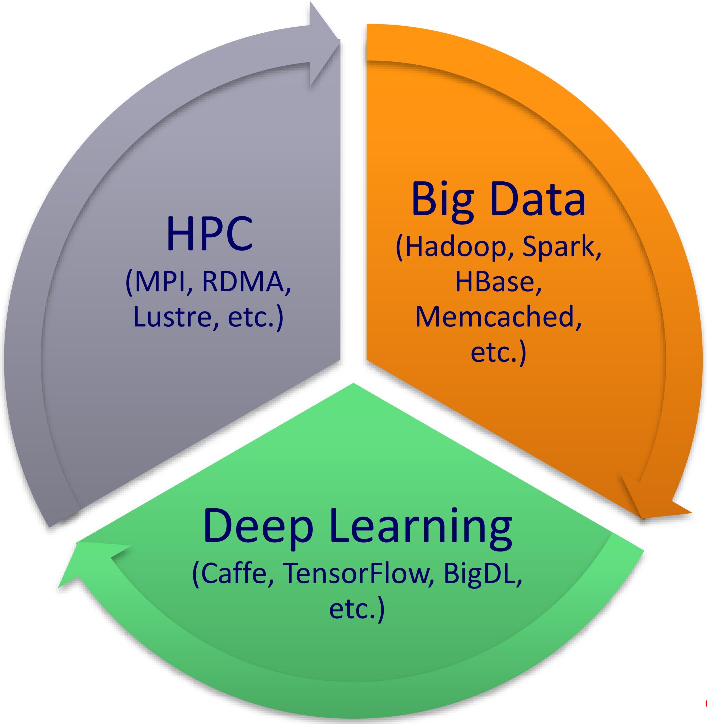
**Dhabaleswar K. (DK) Panda**

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



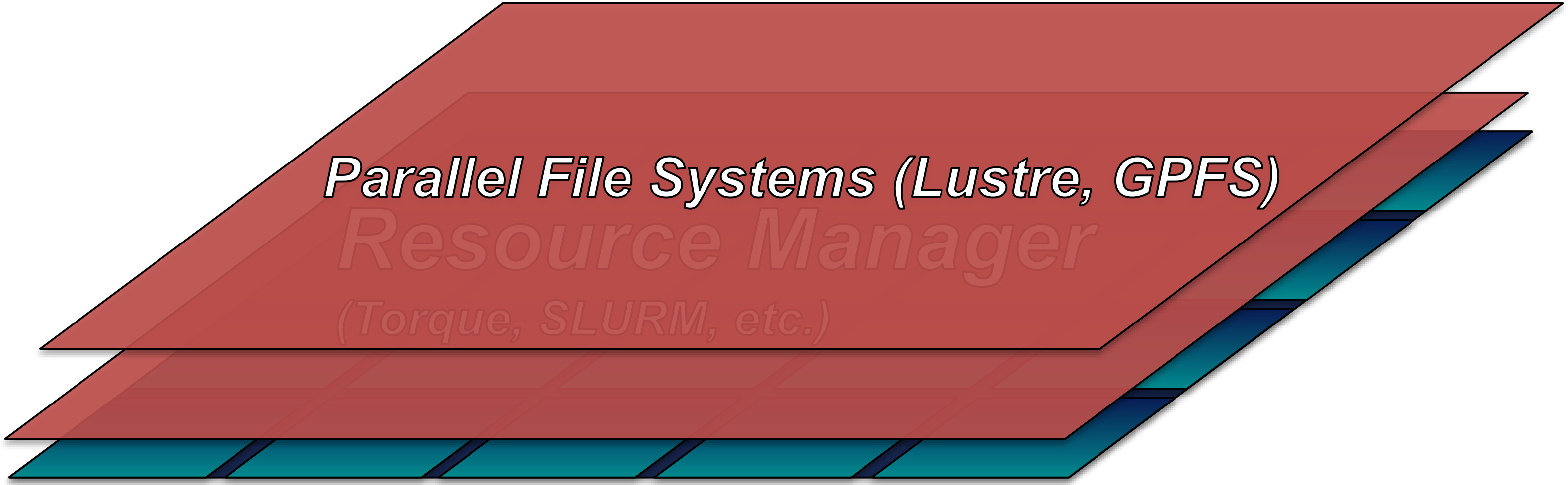

**Convergence of HPC, Big Data, and Deep Learning!**

**Increasing Need to Run these applications on the Cloud!!**

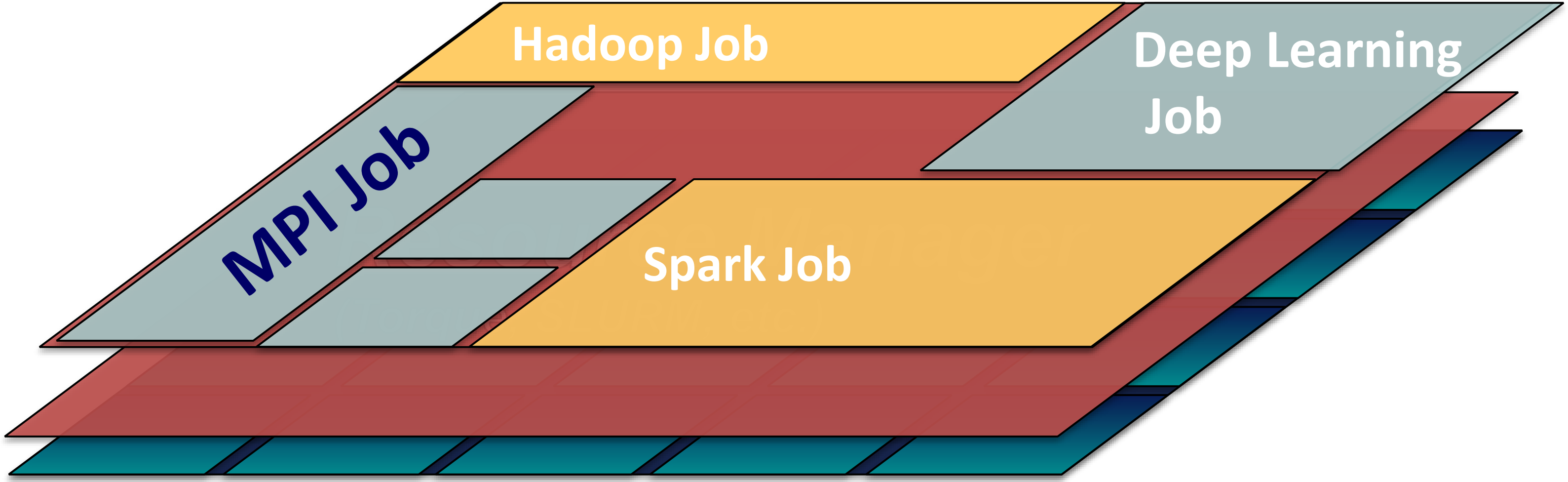
# Can We Run Big Data and Deep Learning Jobs on Existing HPC Infrastructure?

# Can We Run Big Data and Deep Learning Jobs on Existing HPC Infrastructure?

# Can We Run Big Data and Deep Learning Jobs on Existing HPC Infrastructure?



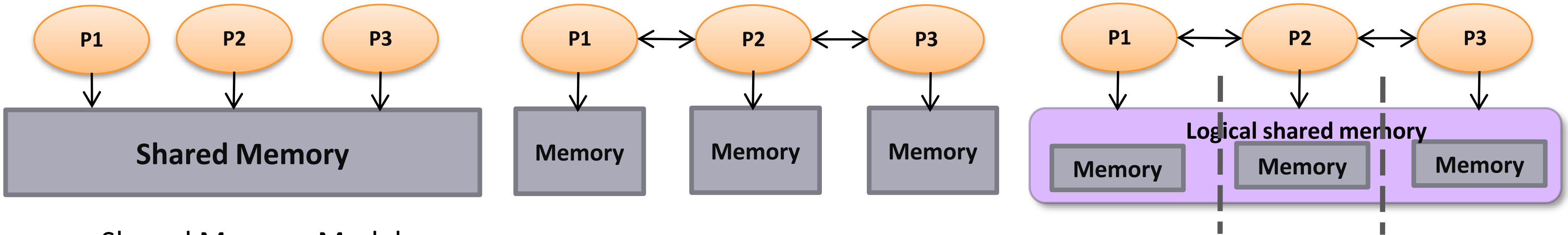
# Can We Run Big Data and Deep Learning Jobs on Existing HPC Infrastructure?



# HPC, Big Data, Deep Learning, and Cloud

- Traditional HPC
  - Message Passing Interface (MPI), including MPI + OpenMP
  - Exploiting Accelerators
- Deep Learning
  - Caffe, CNTK, TensorFlow, and many more
- Big Data/Enterprise/Commercial Computing
  - Spark and Hadoop (HDFS, HBase, MapReduce)
  - Deep Learning over Big Data (DLoBD)
- Cloud for HPC and BigData
  - Virtualization with SR-IOV and Containers

# Parallel Programming Models Overview




Shared Memory Model

SHMEM, DSM

Distributed Memory Model

MPI (Message Passing Interface)

Partitioned Global Address Space (PGAS)

Global Arrays, 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

# Supporting Programming Models for Multi-Petaflop and Exaflop Systems: Challenges

**Application Kernels/Applications**

**Middleware**

**Programming Models**

MPI, PGAS (UPC, Global Arrays, OpenSHMEM), CUDA, OpenMP, OpenACC, Cilk, Hadoop (MapReduce), Spark (RDD, DAG), etc.

**Communication Library or Runtime for Programming Models**

- Point-to-point Communication
- Collective Communication
- Energy-Awareness
- Synchronization and Locks
- I/O and File Systems
- Fault Tolerance

**Networking Technologies**
(InfiniBand, 40/100GigE, Aries, and Omni-Path)

**Multi-/Many-core Architectures**

**Accelerators**
(GPU and FPGA)

Co-Design Opportunities and Challenges across Various Layers

**Performance**

**Scalability**

**Resilience**

# 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

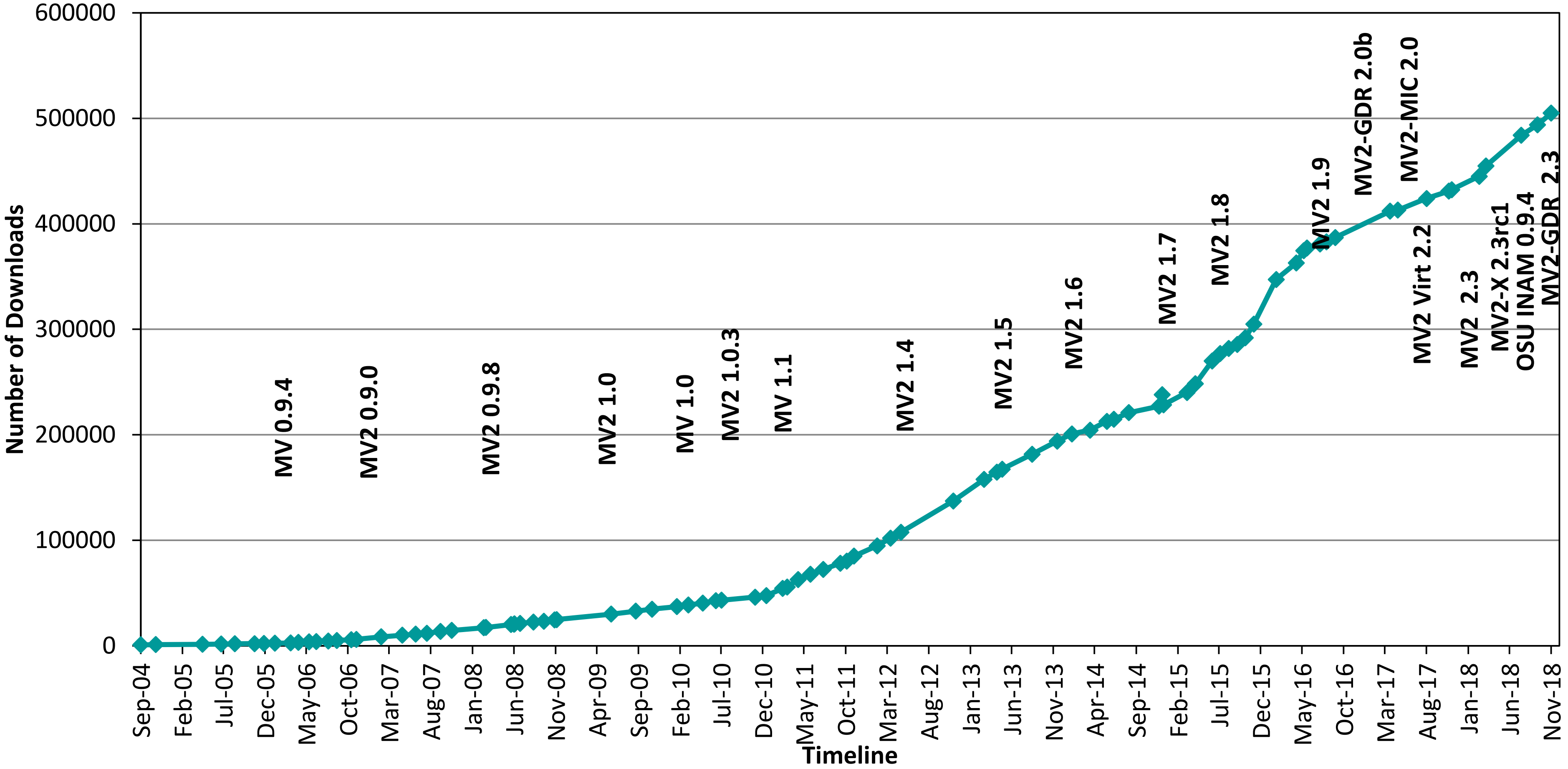
# Overview of the MVAPICH2 Project

- 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 2,950 organizations in 86 countries**
  - **More than 511,000 (> 0.5 million) downloads from the OSU site directly**
  - Empowering many TOP500 clusters (Nov '18 ranking)
    - 3rd ranked 10,649,640-core cluster (Sunway TaihuLight) at NSC, Wuxi, China
    - 14th, 556,104 cores (Oakforest-PACS) in Japan
    - 17th, 367,024 cores (Stampede2) at TACC
    - 27th, 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**


**Partner in the upcoming TACC Frontera System**

- Empowering Top500 systems for over a decade

# 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)

**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

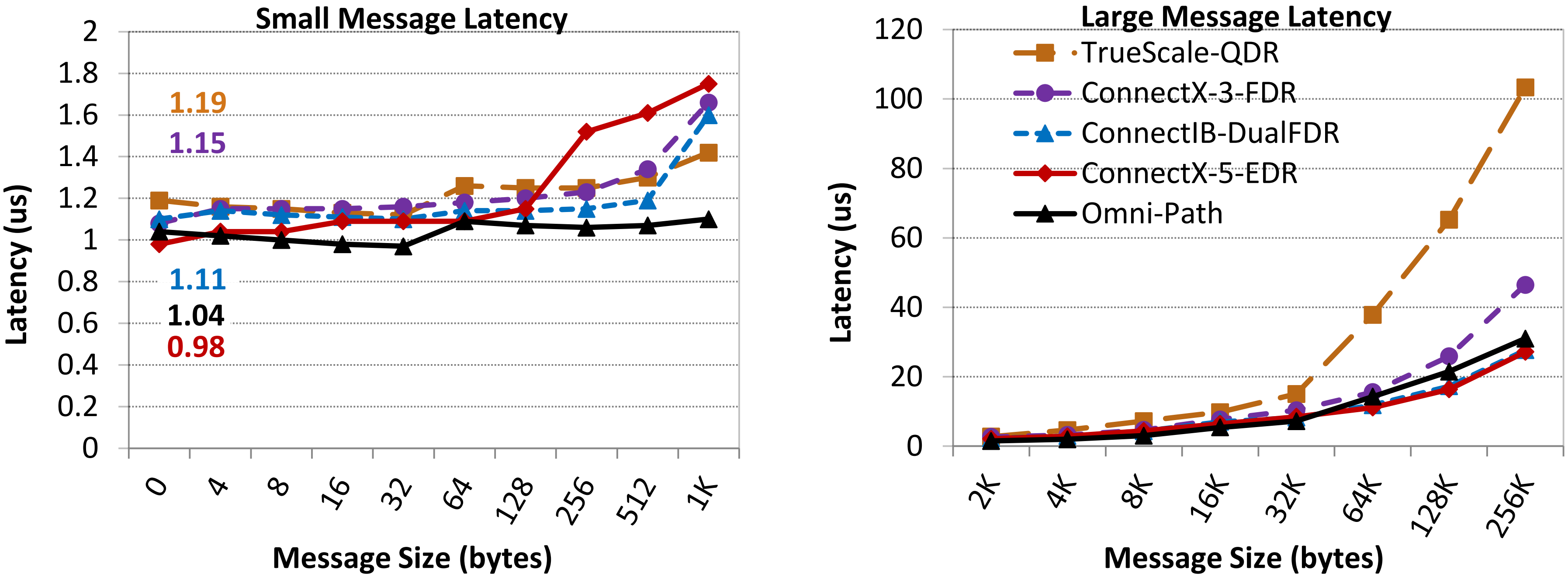
# MVAPICH2 Software Family

| 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 |

# Overview of A Few Challenges being Addressed by the MVAPICH2 Project for Exascale

- Scalability for million to billion processors
  - Support for highly-efficient inter-node and intra-node communication
  - Scalable Start-up
  - Optimized Collectives using SHArP and Multi-Leaders
  - Optimized CMA-based and XPMEM-based Collectives
  - Asynchronous Progress
- Exploiting Accelerators (NVIDIA GPGPUs)
- Optimized MVAPICH2 for OpenPower (with/ NVLink) and ARM
- Application Scalability and Best Practices

# One-way Latency: MPI over IB with MVAPICH2




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

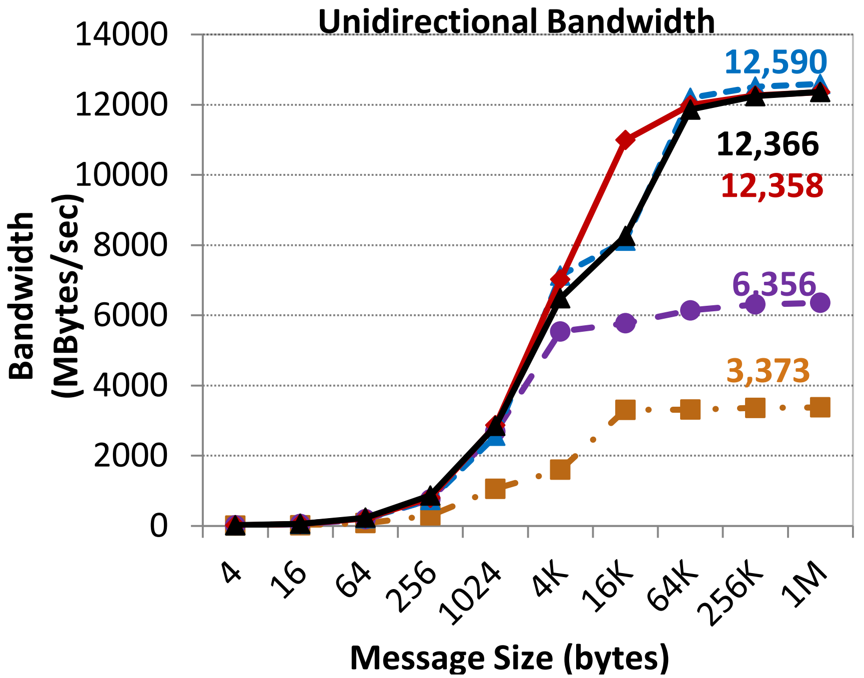
ConnectX-3-FDR - 2.8 GHz Deca-core (IvyBridge) Intel PCI Gen3 with IB switch

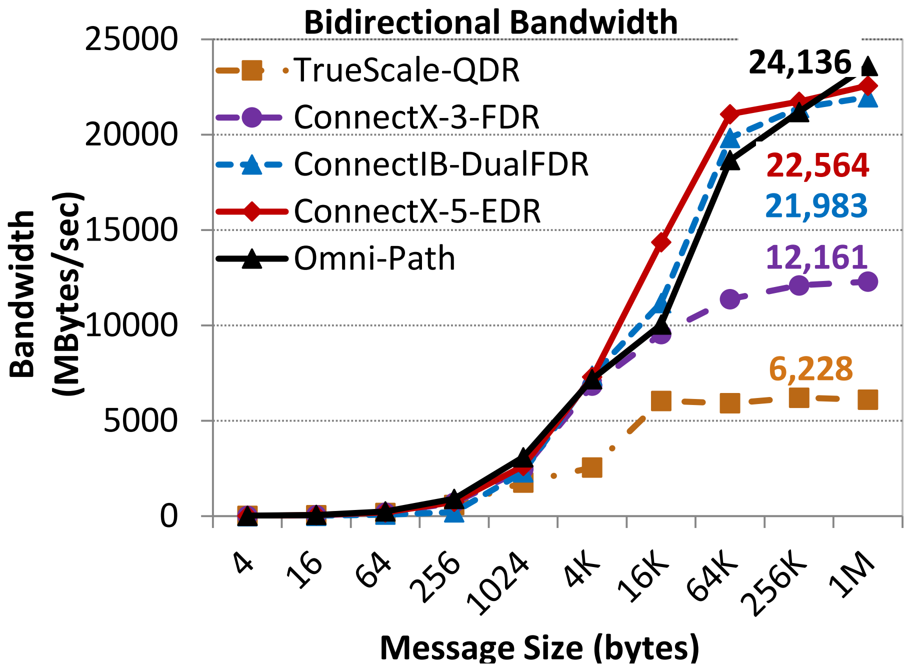
ConnectIB-Dual FDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB switch

ConnectX-5-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

# Bandwidth: MPI over IB with MVAPICH2




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

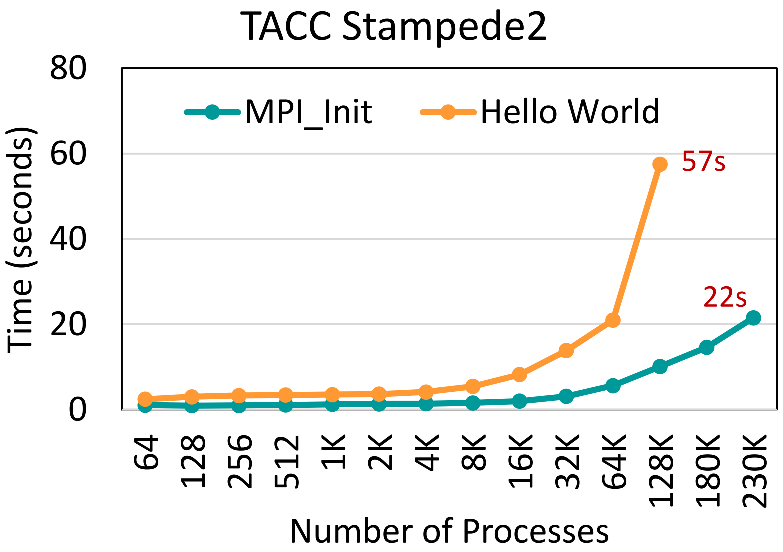
ConnectX-3-FDR - 2.8 GHz Deca-core (IvyBridge) Intel PCI Gen3 with IB switch

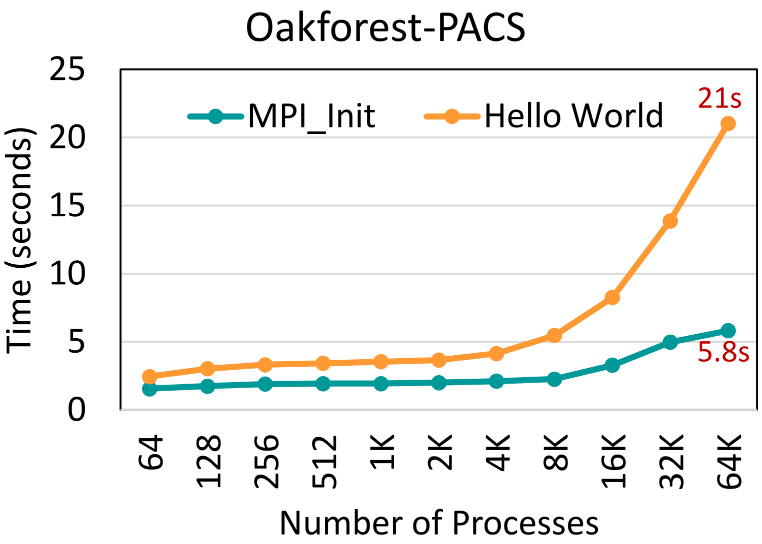
ConnectIB-Dual FDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB switch

ConnectX-5-EDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 IB switch

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

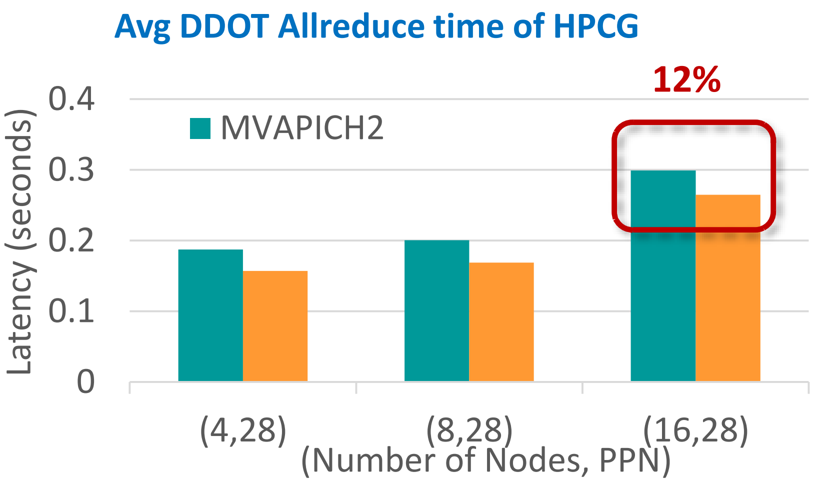
# Startup Performance on KNL + Omni-Path




- MPI_Init takes 22 seconds on 231,936 processes on 3,624 KNL nodes (Stampede2 – Full scale)
- At 64K processes, MPI_Init and Hello World takes 5.8s and 21s respectively (Oakforest-PACS)
- All numbers reported with 64 processes per node, MVAPICH2-2.3a
- Designs integrated with mpirun_rsh, available for srun (SLURM launcher) as well

# Benefits of SHARP Allreduce at Application Level

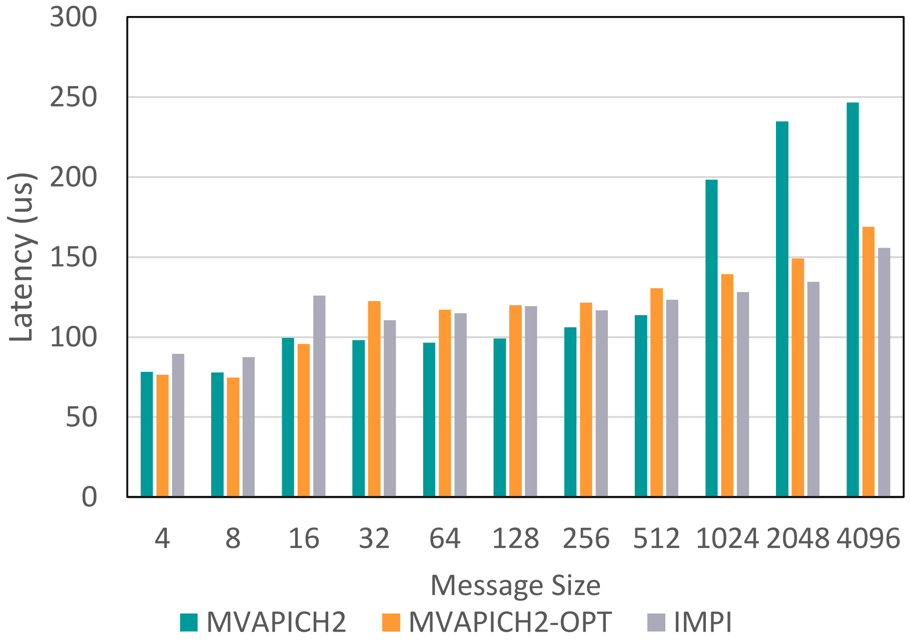


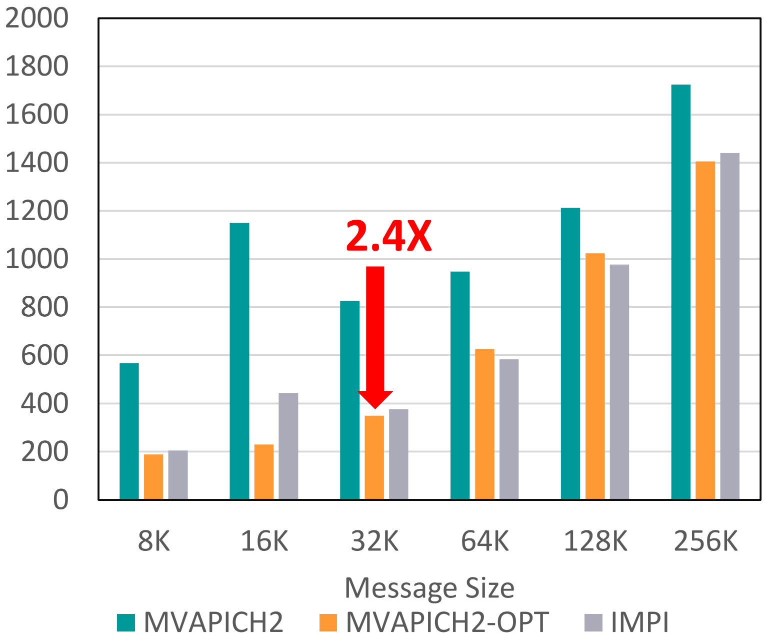

**SHARP support available since MVAPICH2 2.3a**

| Parameter | Description | Default |
|---|---|---|
| MV2_ENABLE_SHARP=1 | Enables SHARP-based collectives | Disabled |
| --enable-sharp | Configure flag to enable SHARP | Disabled |

- **Refer to Running Collectives with Hardware based SHARP support section of MVAPICH2 user guide for more information**
- **http://mvapich.cse.ohio-state.edu/static/media/mvapich/mvapich2-2.3-userguide.html#x1-990006.26**

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



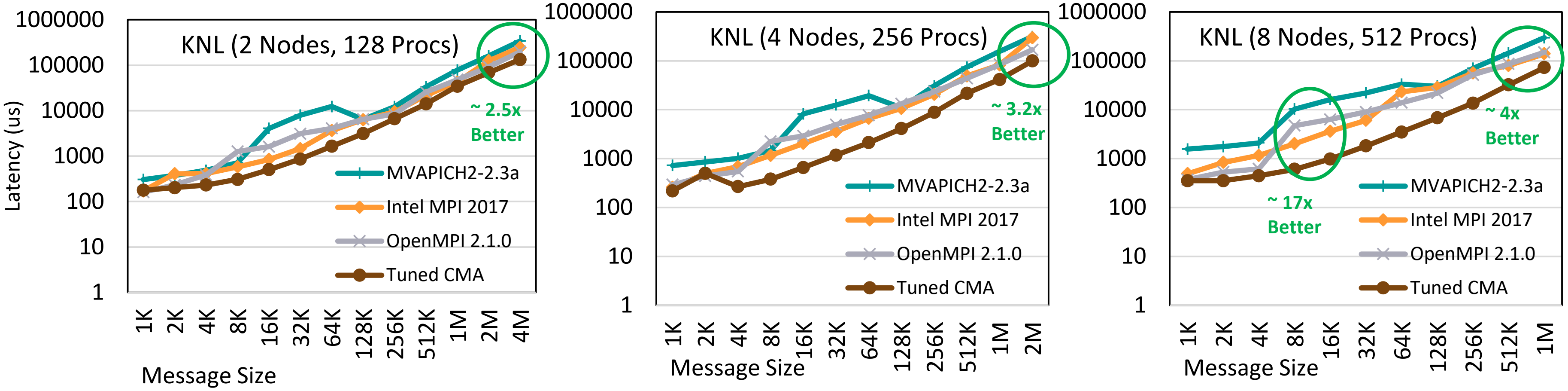

OSU Micro Benchmark 64 PPN

- 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



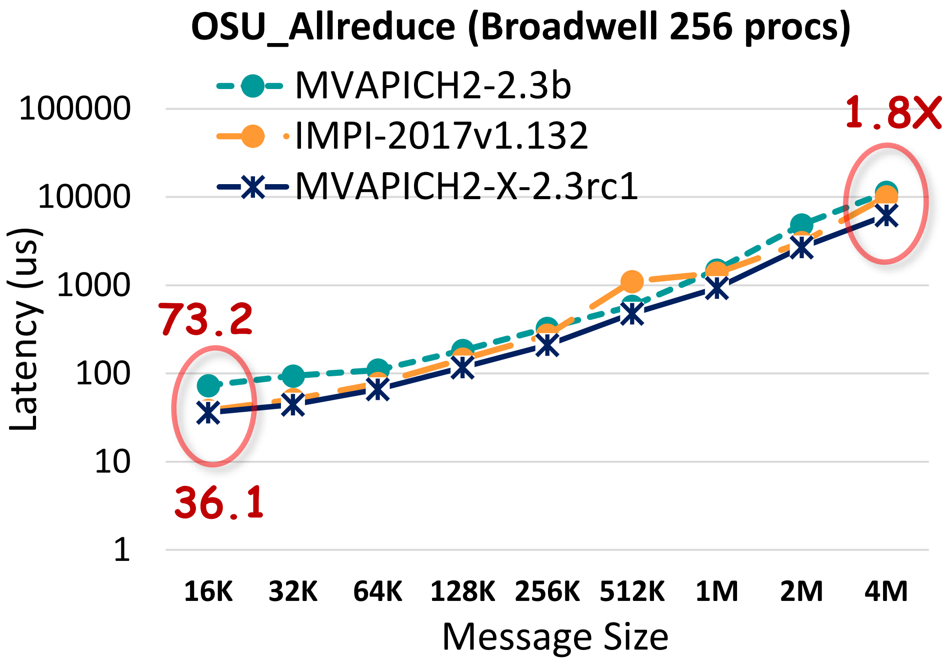

**Performance of MPI_Gather on KNL nodes (64PPN)**

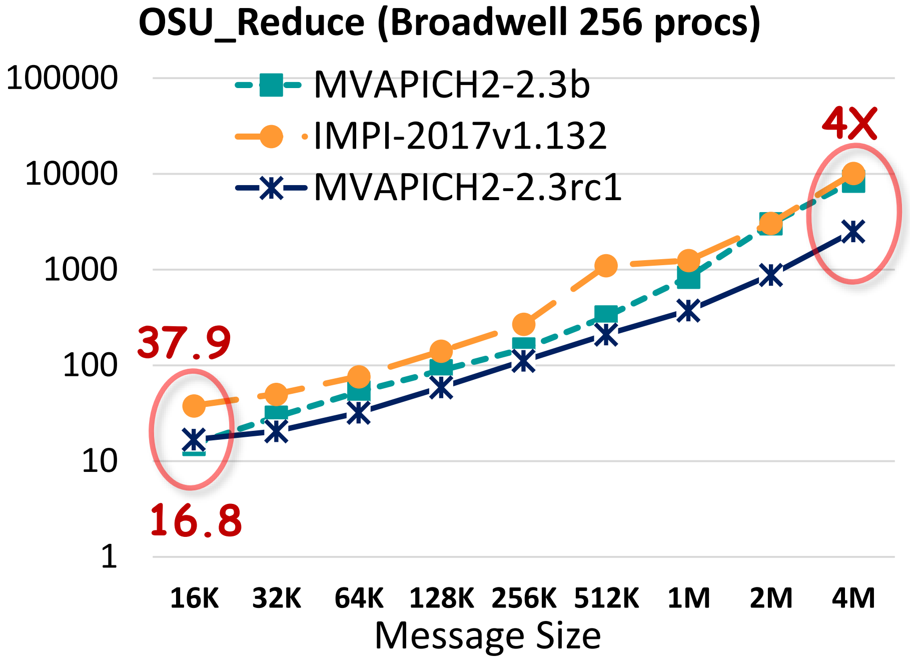
- Significant improvement over existing implementation for Scatter/Gather with 1MB messages (up to 4x on KNL, 2x on Broadwell, 14x on OpenPower)
- New two-level algorithms for better scalability
- Improved performance for other collectives (Bcast, Allgather, and Alltoall)

***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



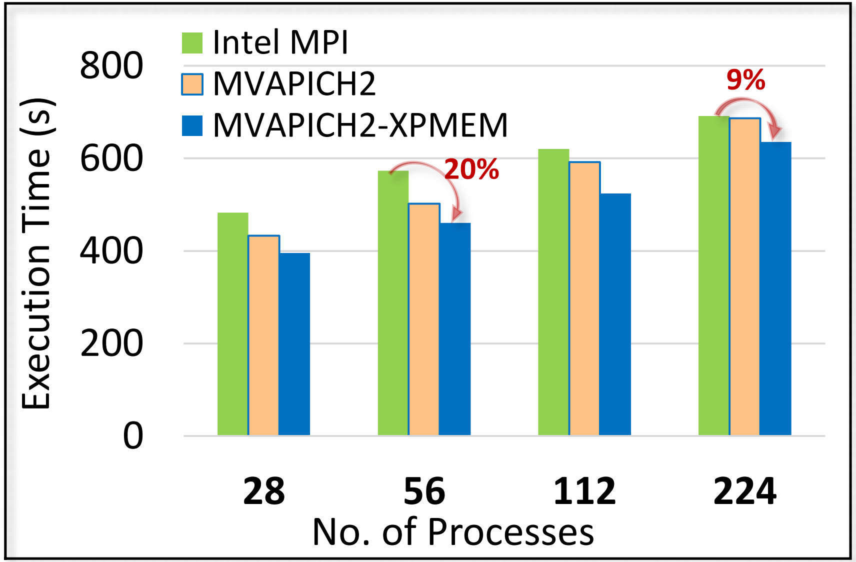

- "*Shared Address Space*"-based true *zero-copy* Reduction collective designs in MVAPICH2
- Offloaded computation/communication to peers ranks in reduction collective operation
- Up to **4X** improvement for 4MB Reduce and up to **1.8X** improvement for 4M AllReduce

*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.*

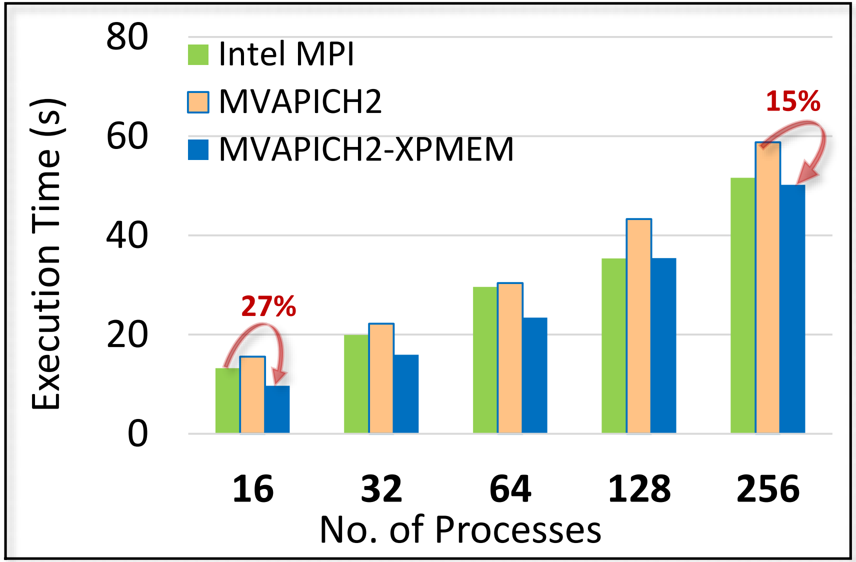
**Available in MVAPICH2-X 2.3rc1**

# Application-Level Benefits of XPMEM-Based Collectives

CNTK AlexNet Training
(Broadwell, B.S=default, iteration=50, ppn=28)

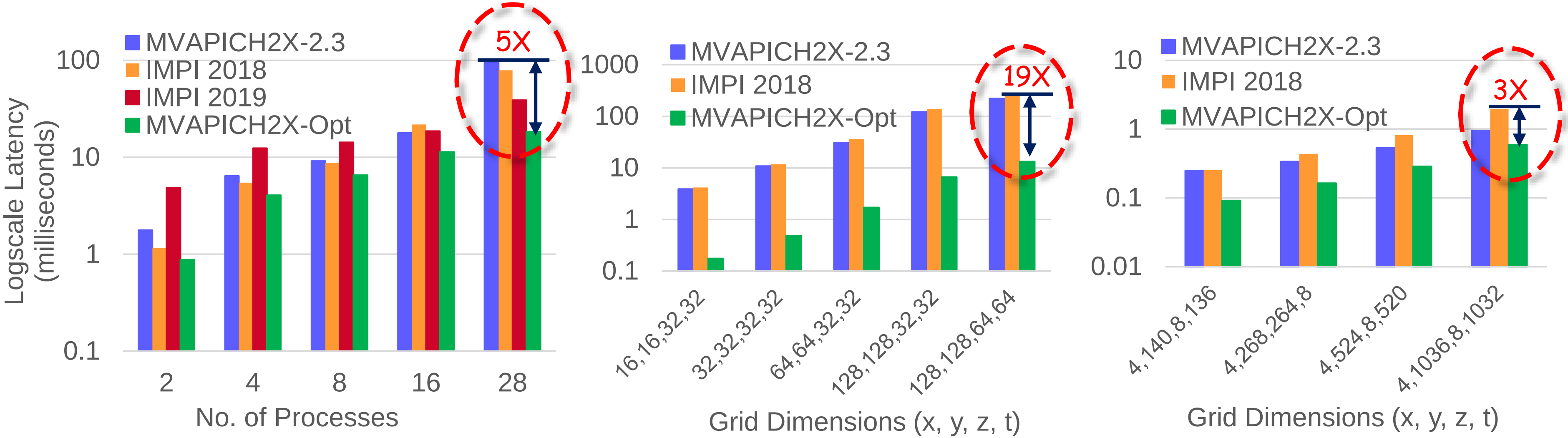



MiniAMR (Broadwell, ppn=16)




- Up to **20%** benefits over IMPI for CNTK DNN training using AllReduce
- Up to **27%** benefits over IMPI and up to **15%** improvement over MVAPICH2 for MiniAMR application kernel

# Efficient Zero-copy MPI Datatypes for Emerging Architectures




**3D-Stencil** Datatype Kernel on **Broadwell** (2x14 core)

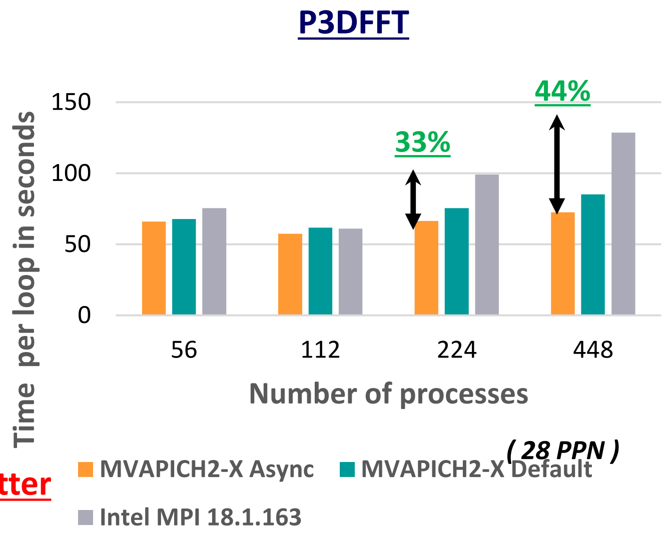
**MILC** Datatype Kernel on **KNL 7250** in Flat-Quadrant Mode (64-core)

**NAS-MG** Datatype Kernel on **OpenPOWER** (20-core)

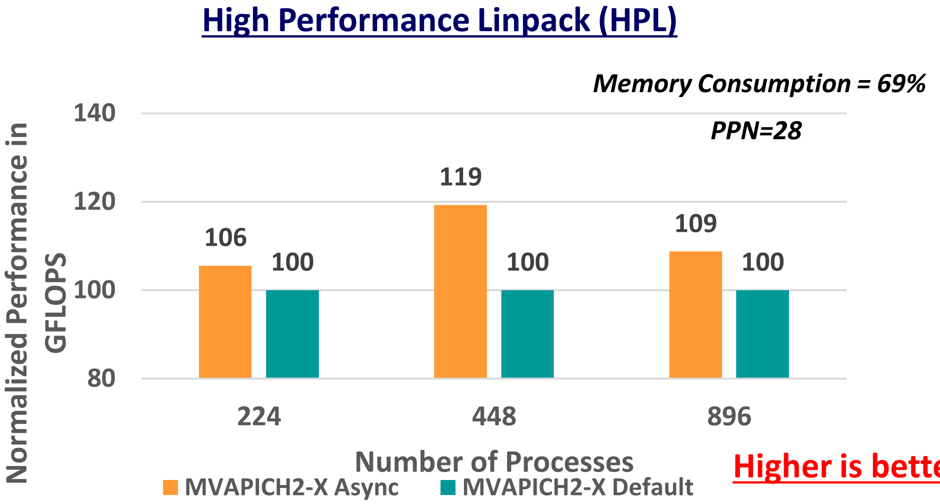
- New designs for efficient zero-copy based MPI derived datatype processing
- Efficient schemes mitigate datatype translation, packing, and exchange overheads
- Demonstrated benefits over prevalent MPI libraries for various application kernels
- To be available in the upcoming MVAPICH2-X release!

**FALCON: Efficient Designs for Zero-copy MPI Datatype Processing on Emerging Architectures**, J. Hashmi, S. Chakraborty, M. Bayatpour, H. Subramoni, D. K. (DK) Panda, 33rd IEEE International Parallel & Distributed Processing Symposium (IPDPS '19), May 2019.

# Benefits of the New Asynchronous Progress Design: Broadwell + InfiniBand




Lower is better




Higher is better

**Up to 44% performance improvement in P3DFFT application with 448 processes**

**Up to 19% and 9% performance improvement in HPL application with 448 and 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

Available in MVAPICH2-X 2.3rc1

# Overview of A Few Challenges being Addressed by the MVAPICH2 Project for Exascale

- Scalability for million to billion processors
- Exploiting Accelerators (NVIDIA GPGPUs)
- Optimized MVAPICH2 for OpenPower (with/ NVLink) and ARM
- Application Scalability and Best Practices

# 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

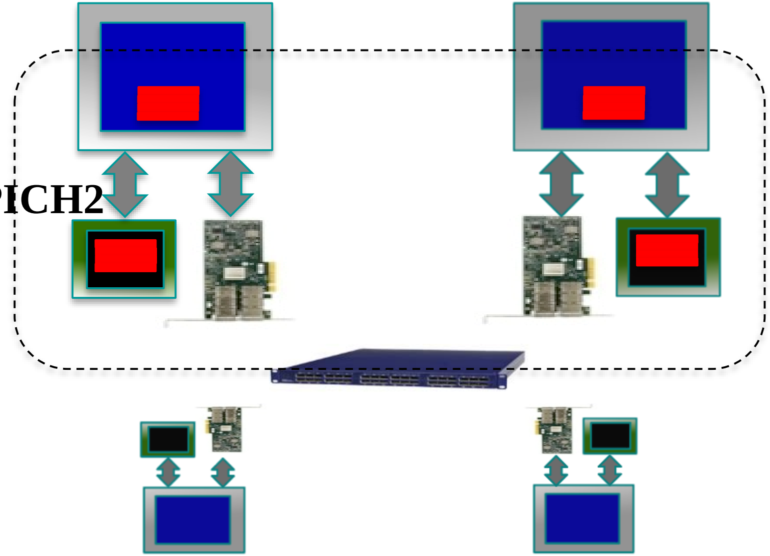
**At Sender:**

```
MPI_Send(s_devbuf, size, …);
```

**At Receiver:**

```
MPI_Recv(r_devbuf, size, …);
```

inside MVAPICH2

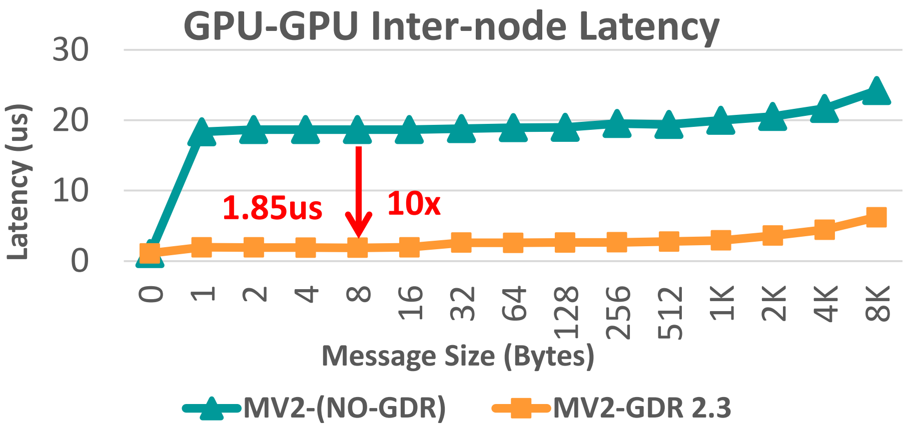


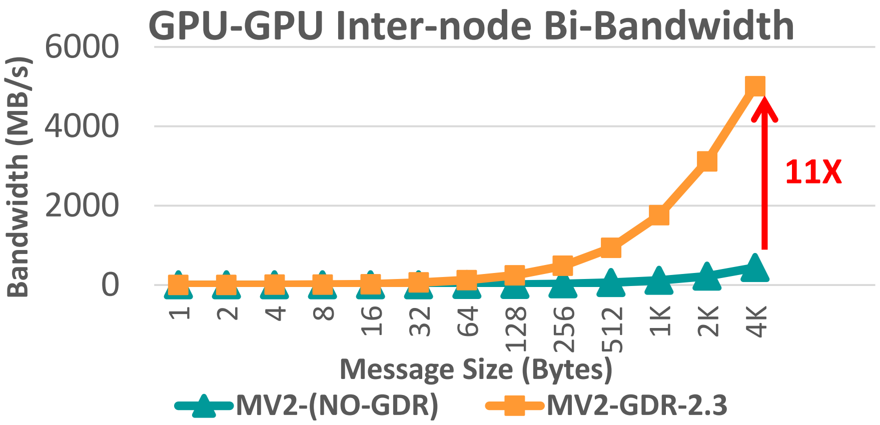
***High Performance and High Productivity***

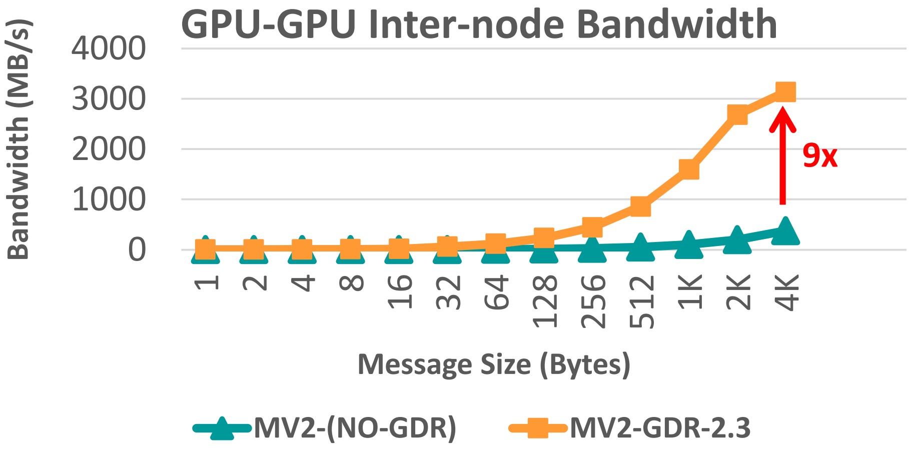
# CUDA-Aware MPI: MVAPICH2-GDR 1.8-2.3 Releases

- Support for MPI communication from NVIDIA GPU device memory
- High performance RDMA-based inter-node point-to-point communication (GPU-GPU, GPU-Host and Host-GPU)
- High performance intra-node point-to-point communication for multi-GPU adapters/node (GPU-GPU, GPU-Host and Host-GPU)
- Taking advantage of CUDA IPC (available since CUDA 4.1) in intra-node communication for multiple GPU adapters/node
- Optimized and tuned collectives for GPU device buffers
- MPI datatype support for point-to-point and collective communication from GPU device buffers
- Unified memory

# Optimized MVAPICH2-GDR Design

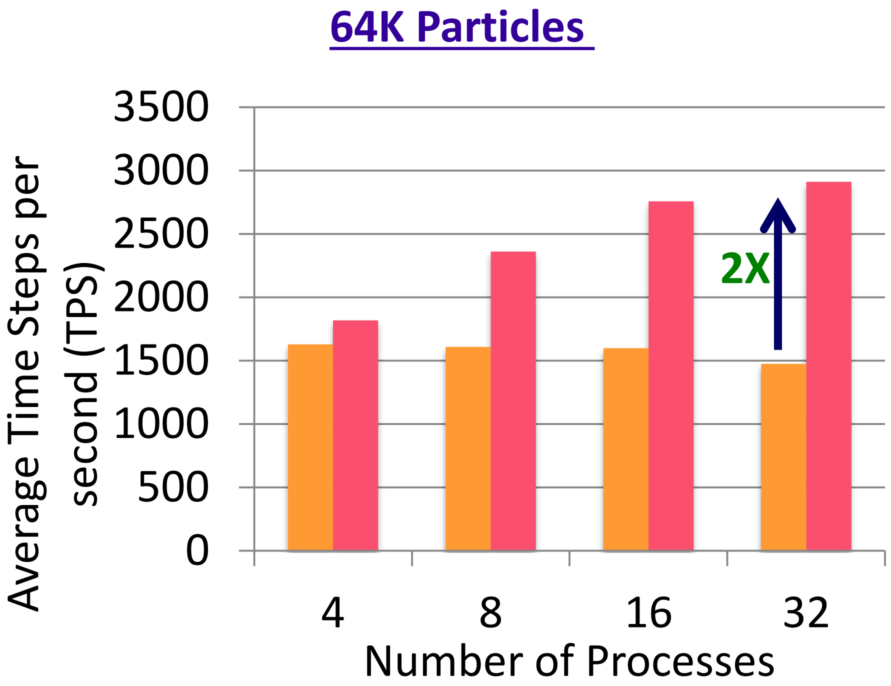


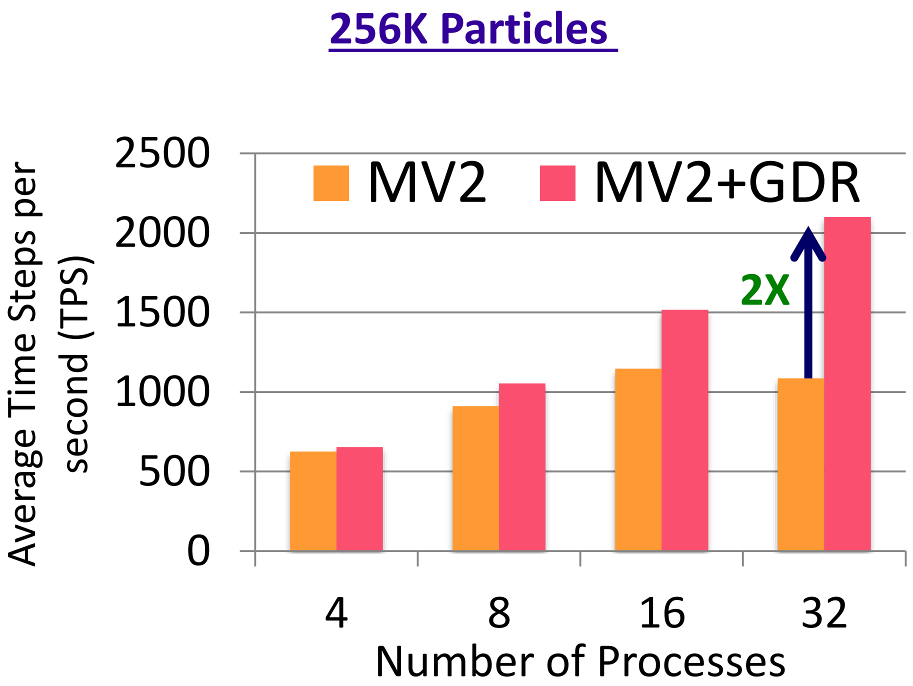




**MVAPICH2-GDR-2.3**
**Intel Haswell (E5-2687W @ 3.10 GHz) node - 20 cores**
**NVIDIA Volta V100 GPU**
**Mellanox Connect-X4 EDR HCA**
**CUDA 9.0**
**Mellanox OFED 4.0 with GPU-Direct-RDMA**

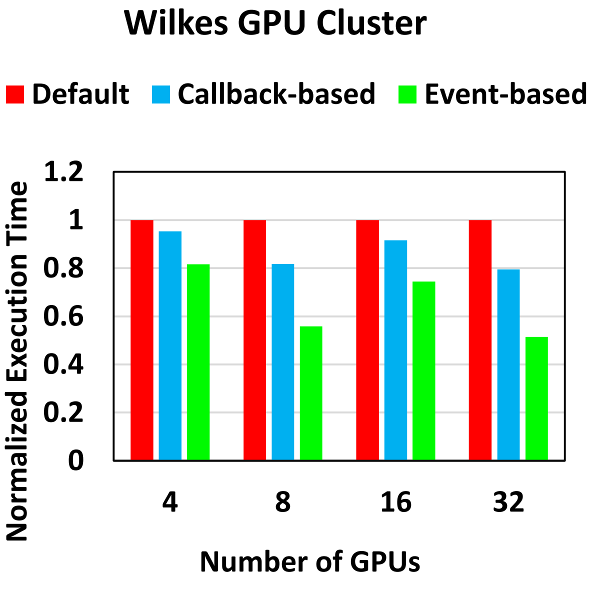
# Application-Level Evaluation (HOOMD-blue)

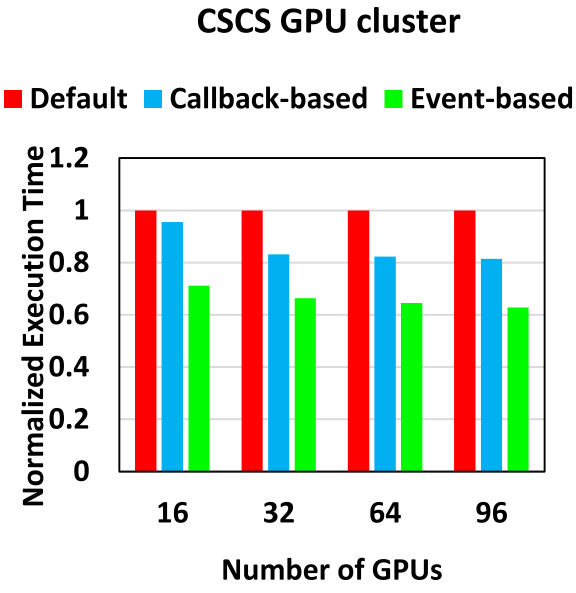


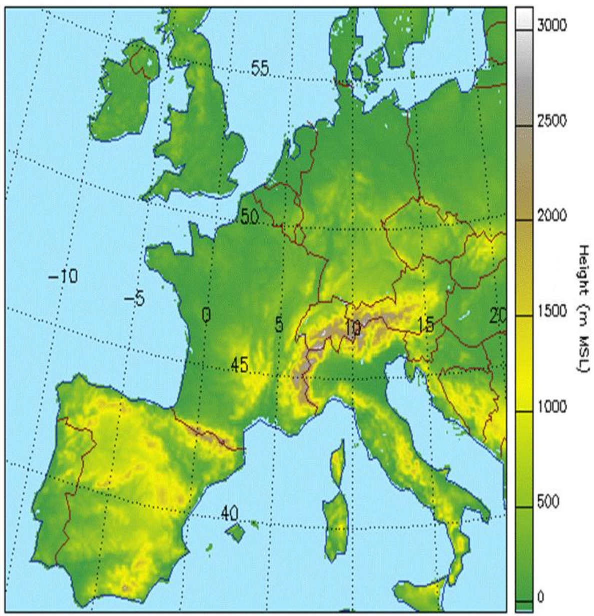




- Platform: Wilkes (Intel Ivy Bridge + NVIDIA Tesla K20c + Mellanox Connect-IB)
- HoomdBlue Version 1.0.5
  - GDRCOPY enabled: MV2_USE_CUDA=1 MV2_IBA_HCA=mlx5_0 MV2_IBA_EAGER_THRESHOLD=32768 MV2_VBUF_TOTAL_SIZE=32768 MV2_USE_GPUDIRECT_LOOPBACK_LIMIT=32768 MV2_USE_GPUDIRECT_GDRCOPY=1 MV2_USE_GPUDIRECT_GDRCOPY_LIMIT=16384

# 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/

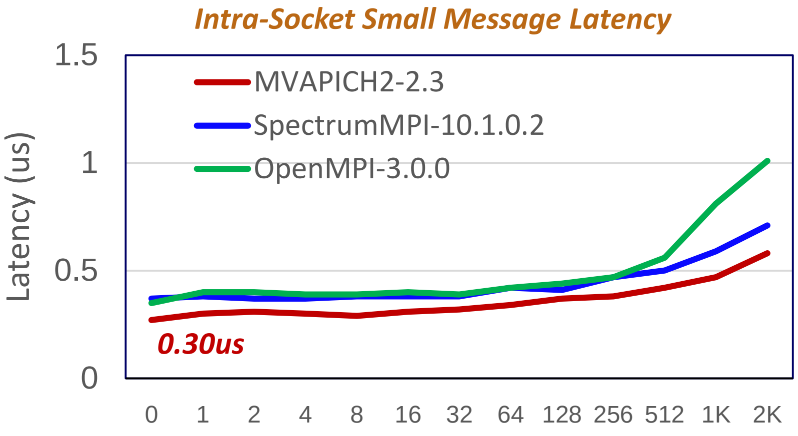
**On-going collaboration with CSCS and MeteoSwiss (Switzerland) in co-designing MV2-GDR and Cosmo Application**

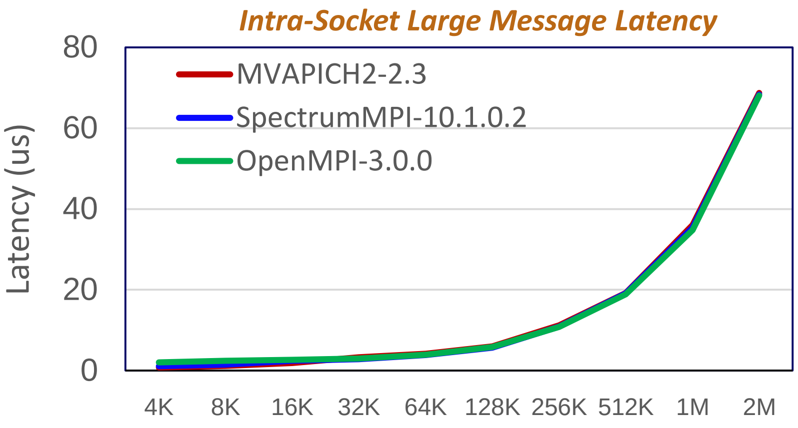
**C. Chu, K. Hamidouche, A. Venkatesh, D. Banerjee , H. Subramoni, and D. K. Panda, Exploiting Maximal Overlap for Non-Contiguous Data Movement Processing on Modern GPU-enabled Systems, IPDPS'16**

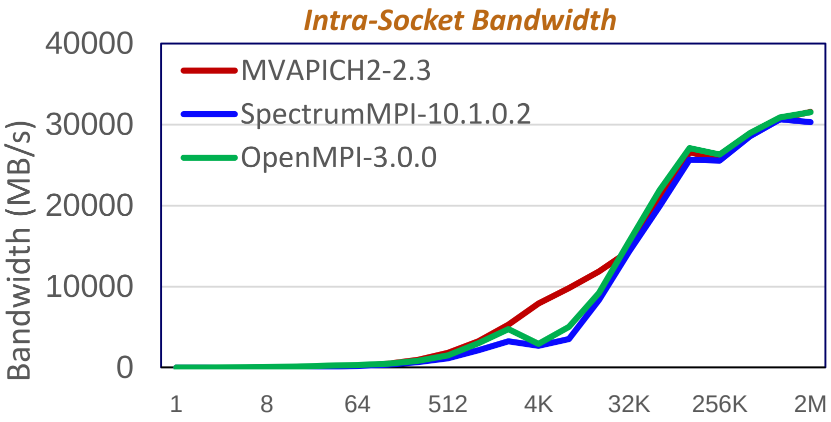
# Overview of A Few Challenges being Addressed by the MVAPICH2 Project for Exascale

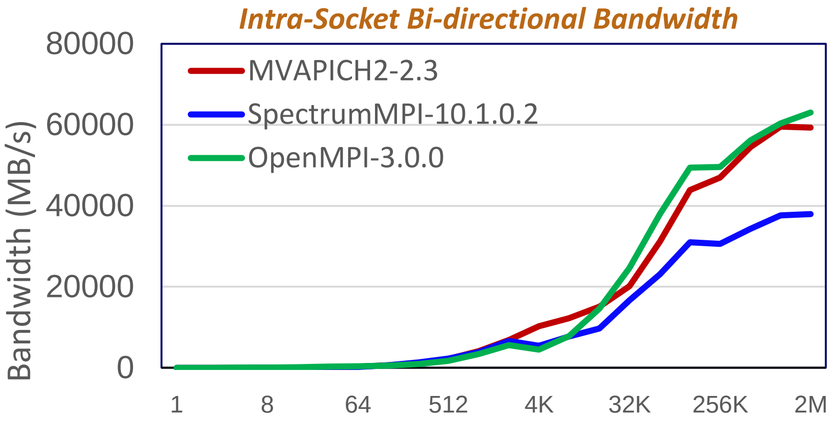
- Scalability for million to billion processors
- Exploiting Accelerators (NVIDIA GPGPUs)
- Optimized MVAPICH2 for OpenPower (with/ NVLink) and ARM
- Application Scalability and Best Practices

# Intra-node Point-to-Point Performance on OpenPower



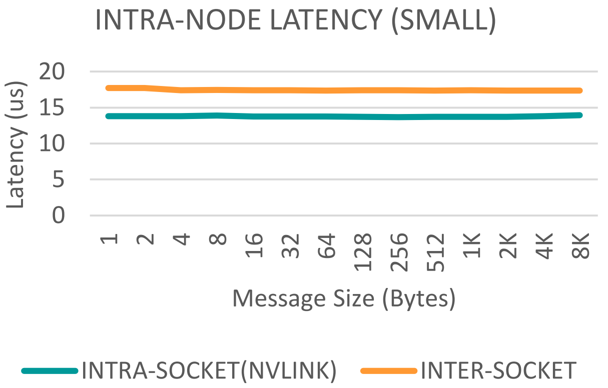



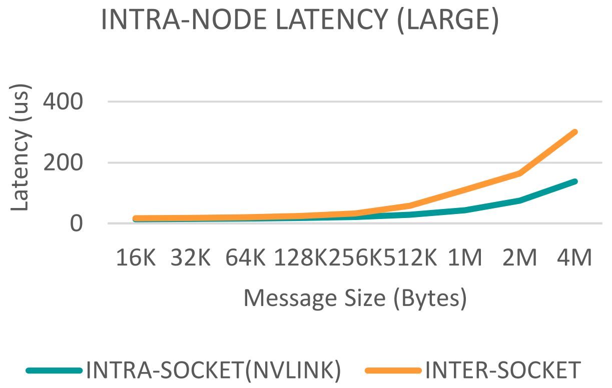



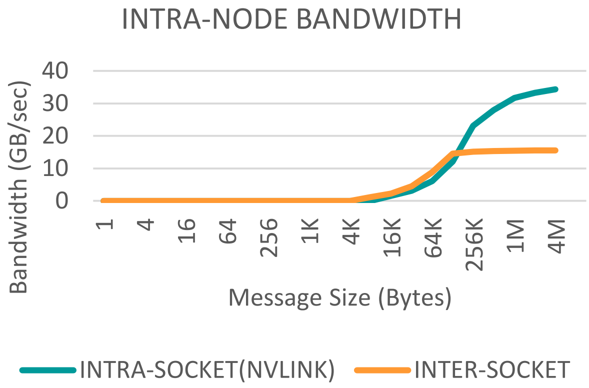




*Platform: Two nodes of OpenPOWER (Power8-ppc64le) CPU using Mellanox EDR (MT4115) HCA*

# MVAPICH2-GDR: Performance on OpenPOWER (NVLink + Pascal)

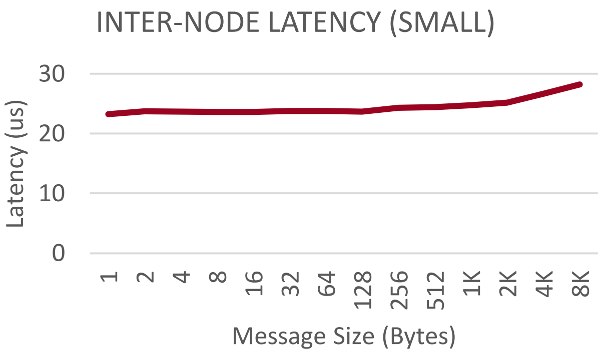


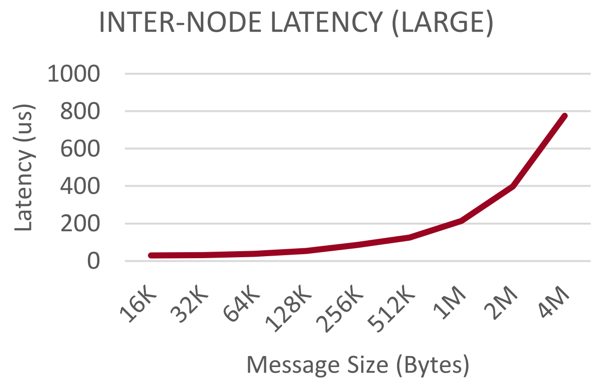



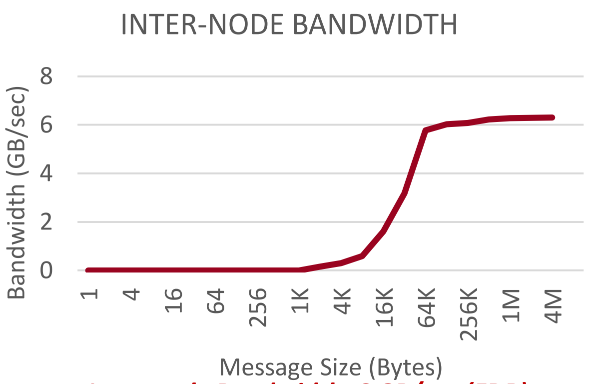




***Intra-node Latency: 13.8 us (without GPUDirectRDMA)***

***Intra-node Bandwidth: 33.2 GB/sec (NVLINK)***



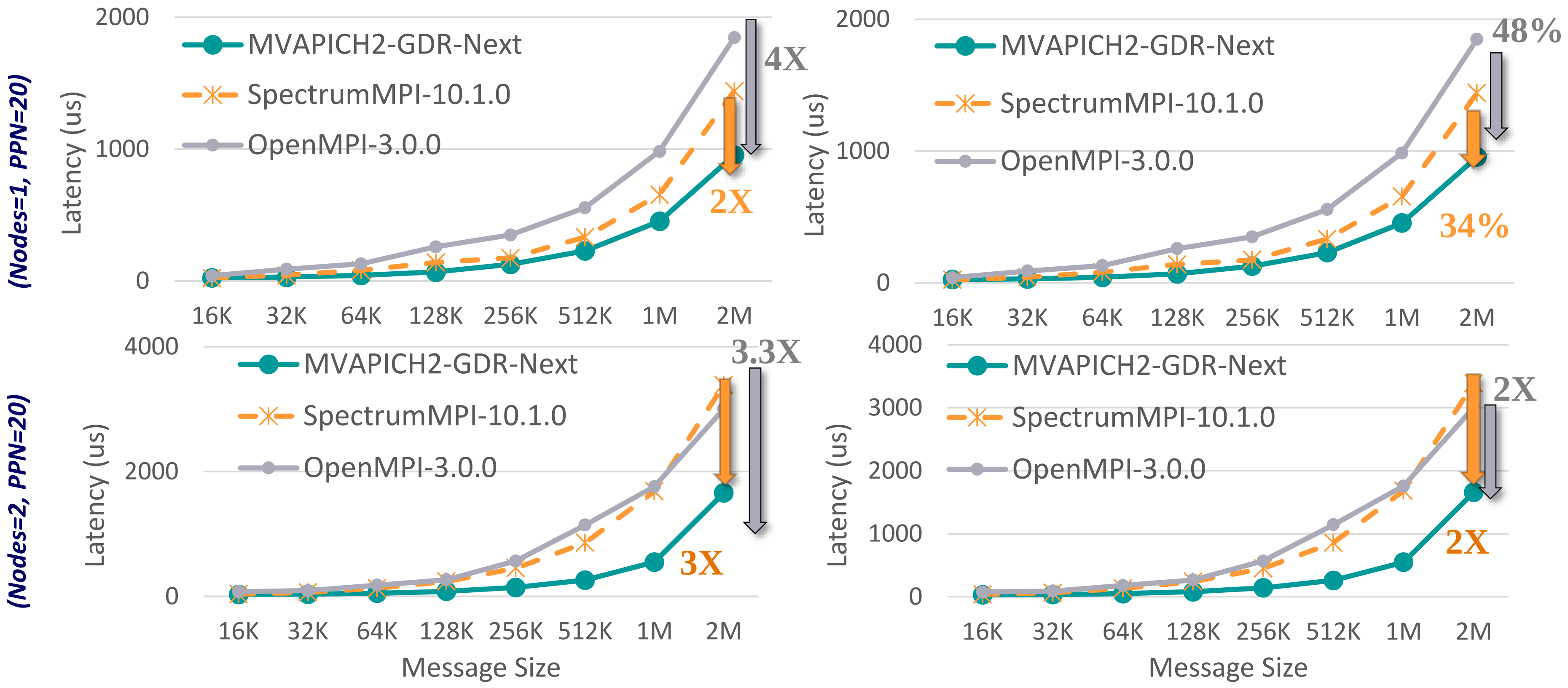

***Inter-node Latency: 23 us (without GPUDirectRDMA)***

Available since MVAPICH2-GDR 2.3a

***Inter-node Bandwidth: 6 GB/sec (FDR)***

*Platform: OpenPOWER (ppc64le) nodes equipped with a dual-socket CPU, 4 Pascal P100-SXM GPUs, and 4X-FDR InfiniBand Inter-connect*

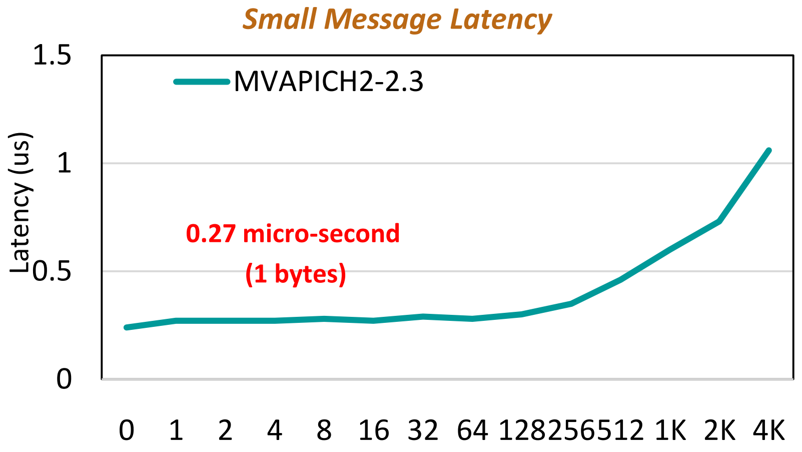
# Optimized All-Reduce with XPMEM on OpenPOWER

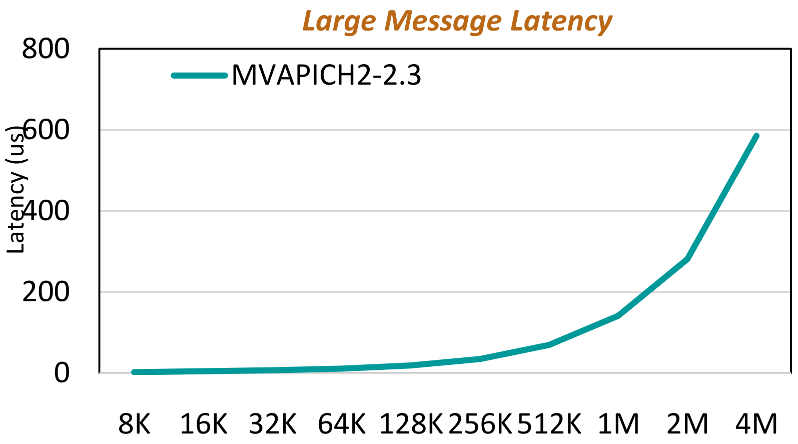


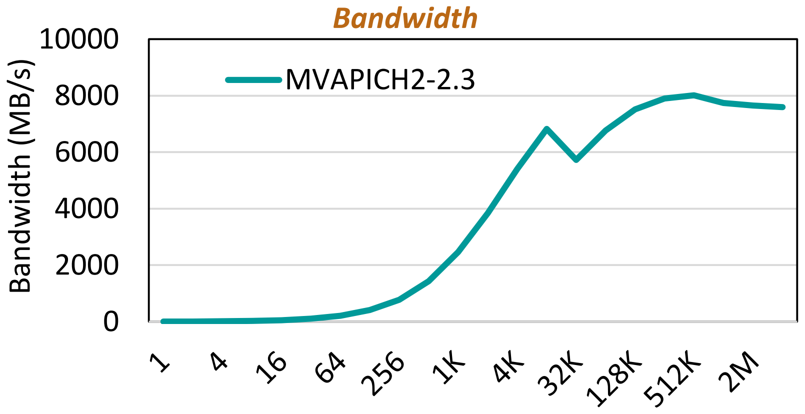

- **Optimized MPI All-Reduce Design in MVAPICH2**
  - *Up to 2X* performance improvement over Spectrum MPI and 4X over OpenMPI for intra-node

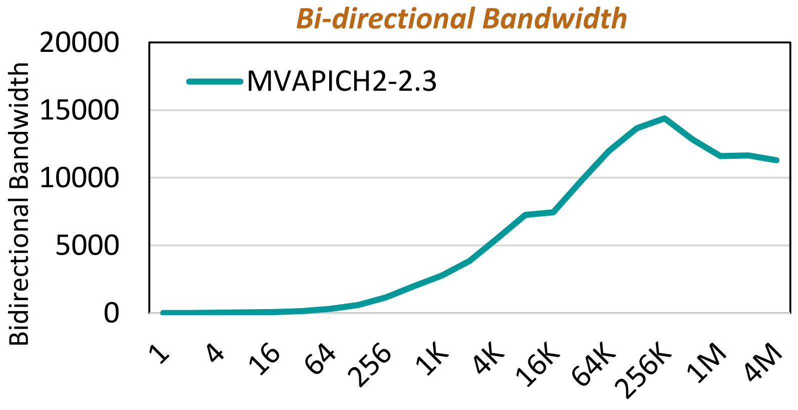
*Optimized Runtime Parameters: MV2_CPU_BINDING_POLICY=hybrid MV2_HYBRID_BINDING_POLICY=bunch*

# Intra-node Point-to-point Performance on ARM Cortex-A72



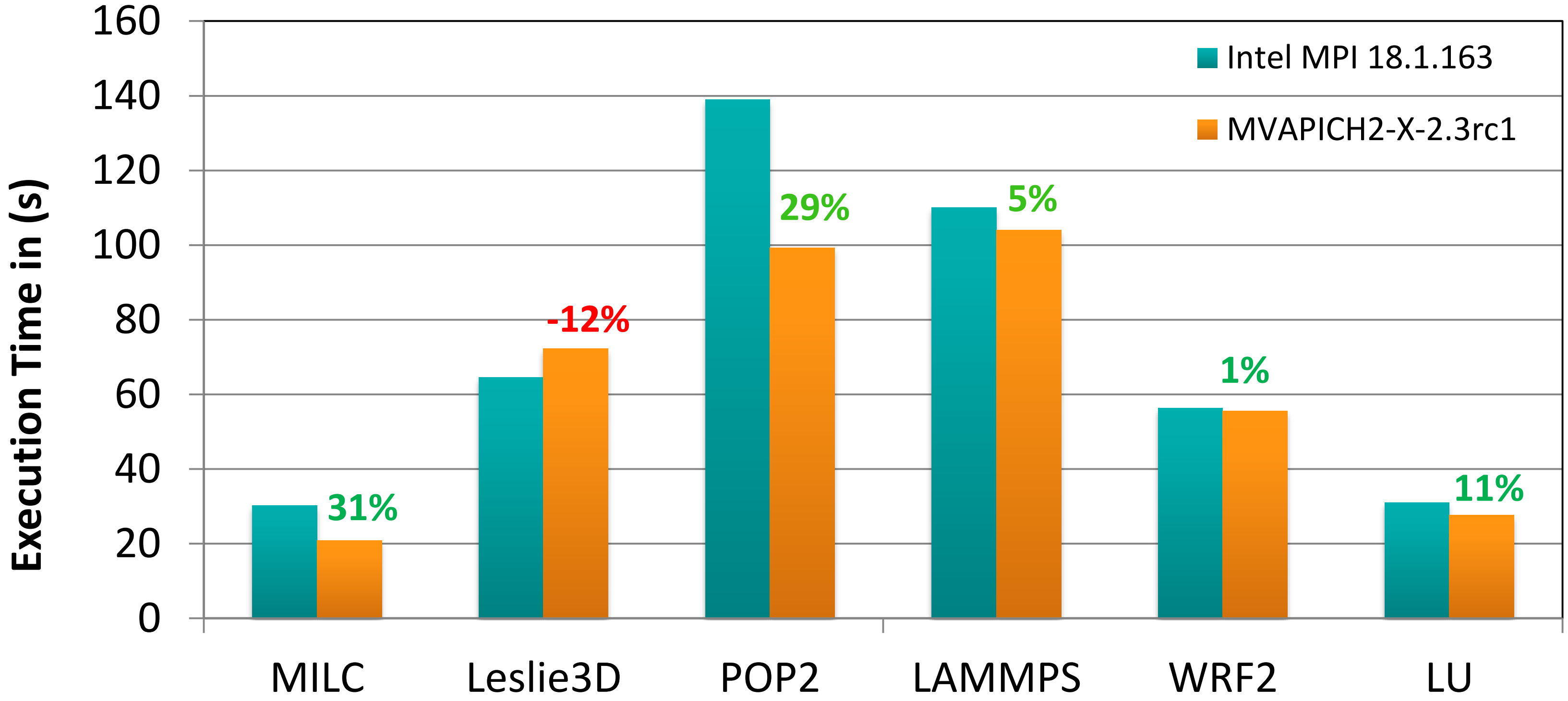




***Platform: ARM Cortex A72 (aarch64) processor with 64 cores dual-socket CPU. Each socket contains 32 cores.***

# Overview of A Few Challenges being Addressed by the MVAPICH2 Project for Exascale

- Scalability for million to billion processors
- Exploiting Accelerators (NVIDIA GPGPUs)
- Optimized MVAPICH2 for OpenPower (with/ NVLink) and ARM
- Application Scalability and Best Practices

# SPEC MPI 2007 Benchmarks: Broadwell + InfiniBand

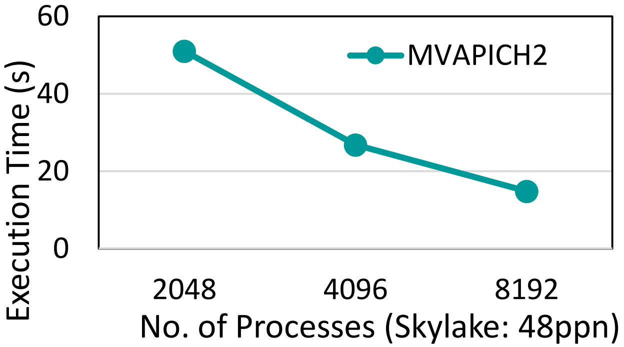


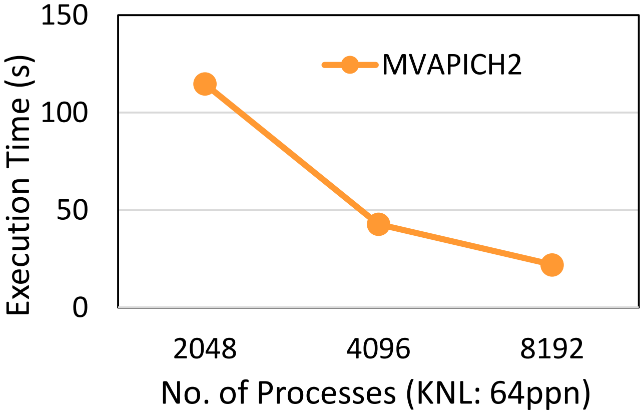

**MVAPICH2-X outperforms Intel MPI by up to 31%**

**Configuration: 448 processes on 16 Intel E5-2680v4 (Broadwell) nodes having 28 PPN and interconnected with 100Gbps Mellanox MT4115 EDR ConnectX-4 HCA**

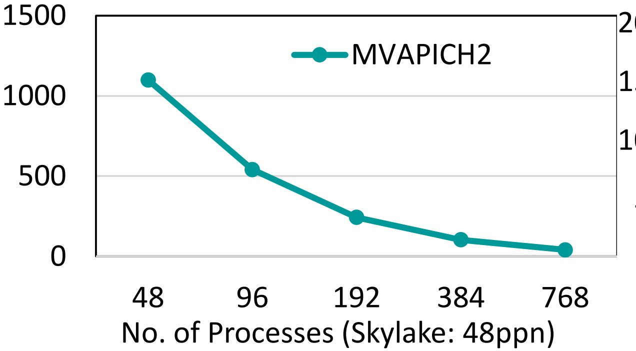
# Application Scalability on Skylake and KNL (Stamepede2)

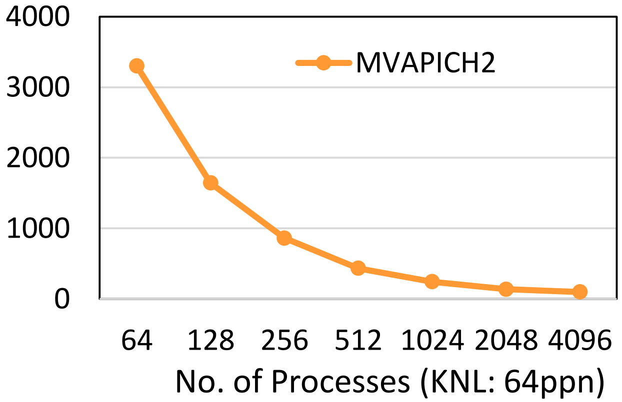
**MiniFE** (1300x1300x1300 ~ 910 GB)



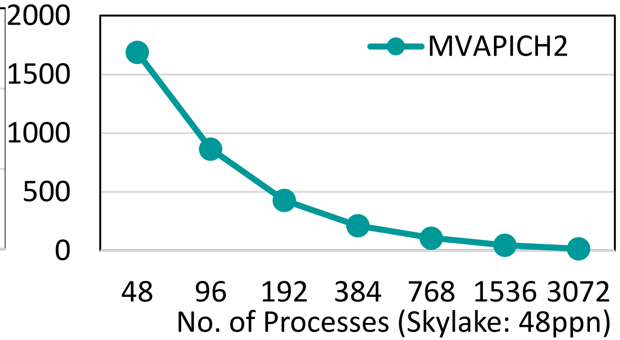



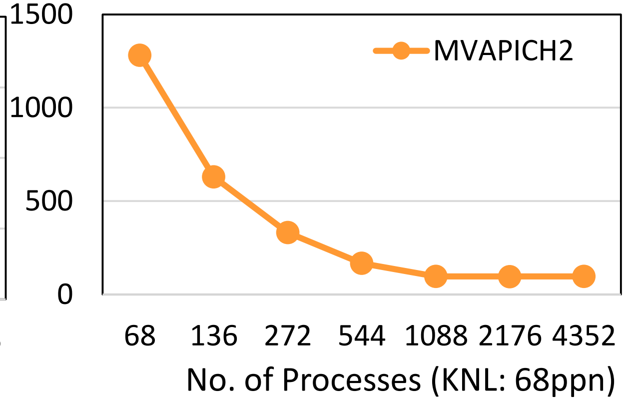

**NEURON** (YuEtAl2012)




**Cloverleaf** (bm64) MPI+OpenMP, NUM_OMP_THREADS = 2




*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*

# Applications-Level Tuning: Compilation of Best Practices

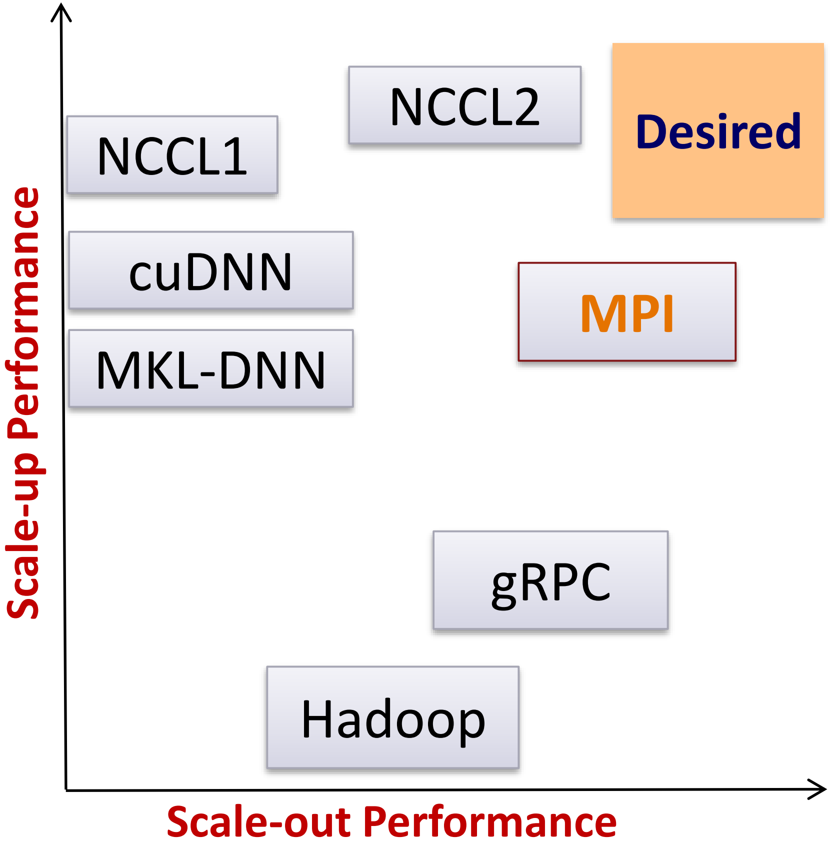
- MPI runtime has many parameters
- Tuning a set of parameters can help you to extract higher performance
- Compiled a list of such contributions through the MVAPICH Website
  - http://mvapich.cse.ohio-state.edu/best_practices/
- Initial list of applications
  - Amber
  - HoomDBlue
  - HPCG
  - Lulesh
  - MILC
  - Neuron
  - SMG2000
  - Cloverleaf
  - SPEC (LAMMPS, POP2, TERA_TF, WRF2)
- Soliciting additional contributions, send your results to mvapich-help at cse.ohio-state.edu.
- We will link these results with credits to you.

# HPC, Big Data, Deep Learning, and Cloud

- Traditional HPC
  - Message Passing Interface (MPI), including MPI + OpenMP
  - Exploiting Accelerators
- Deep Learning
  - Caffe, CNTK, TensorFlow, and many more
- Big Data/Enterprise/Commercial Computing
  - Spark and Hadoop (HDFS, HBase, MapReduce)
  - Deep Learning over Big Data (DLoBD)
- Cloud for HPC and BigData
  - Virtualization with SR-IOV and Containers

# Deep Learning: New Challenges for MPI Runtimes

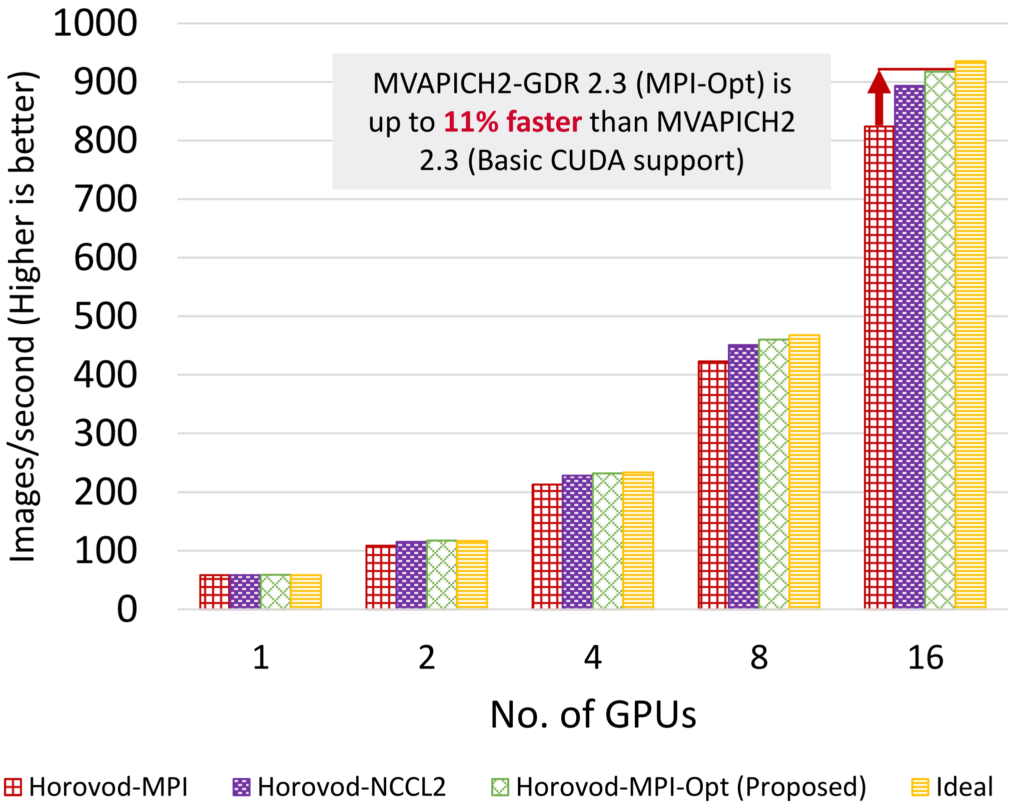
- Deep Learning frameworks are a different game altogether
  - Unusually large message sizes (order of megabytes)
  - Most communication based on GPU buffers
- Existing State-of-the-art
  - cuDNN, cuBLAS, NCCL --> **scale-up** performance
  - NCCL2, CUDA-Aware MPI --> **scale-out** performance
    - For small and medium message sizes only!
- Proposed: Can we **co-design** the MPI runtime (**MVAPICH2-GDR**) and the DL framework (**Caffe**) to achieve both?
  - 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)

# Exploiting CUDA-Aware MPI for TensorFlow (Horovod)

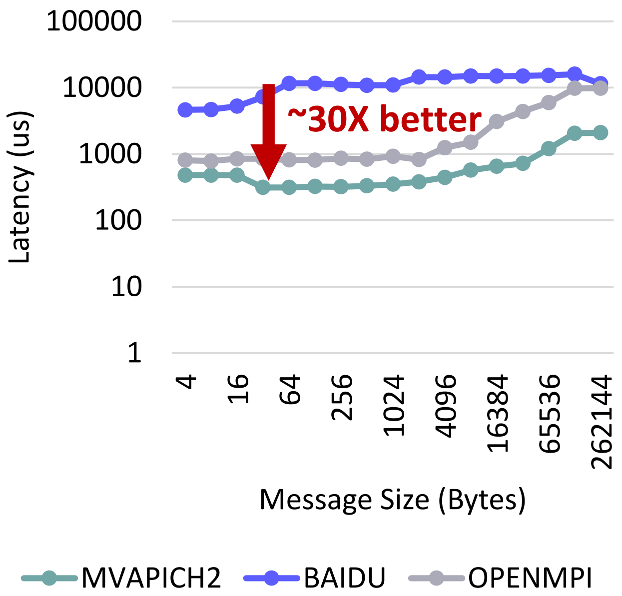
- MVAPICH2-GDR offers excellent performance via advanced designs for MPI_Allreduce.
- Up to **11% better** performance on the RI2 cluster (16 GPUs)
- Near-ideal – **98% scaling efficiency**

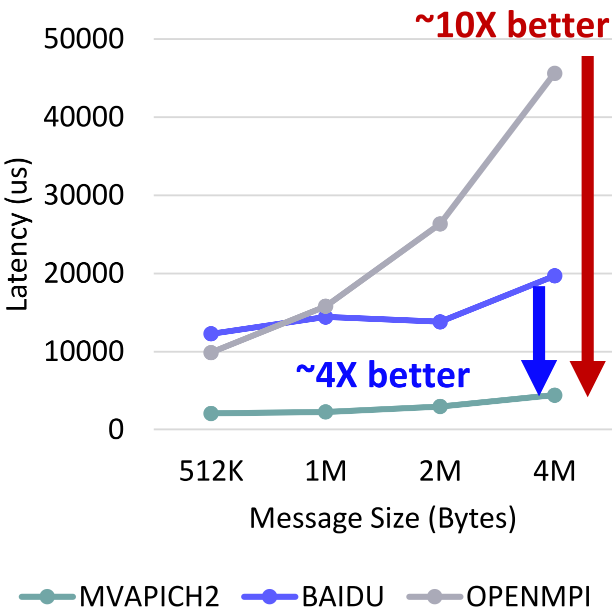


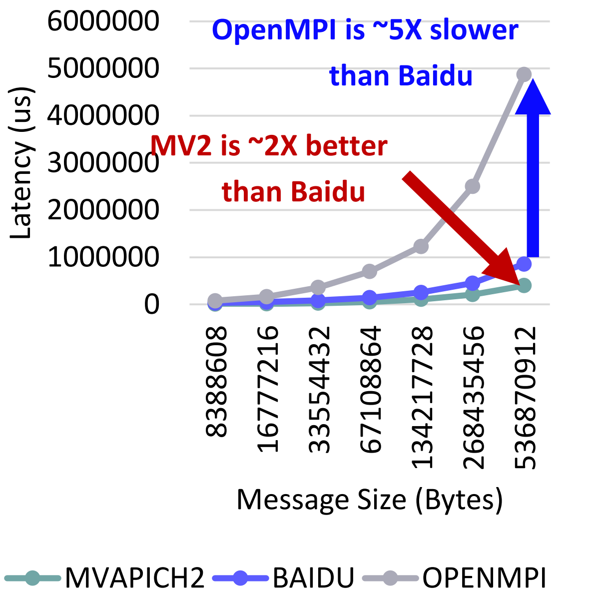

A. A. Awan et al., "Scalable Distributed DNN Training using TensorFlow and CUDA-Aware MPI: Characterization, Designs, and Performance Evaluation", Under Review, https://arxiv.org/abs/1810.11112

# MVAPICH2-GDR: Allreduce Comparison with Baidu and OpenMPI

- 16 GPUs (4 nodes) MVAPICH2-GDR vs. Baidu-Allreduce and OpenMPI 3.0



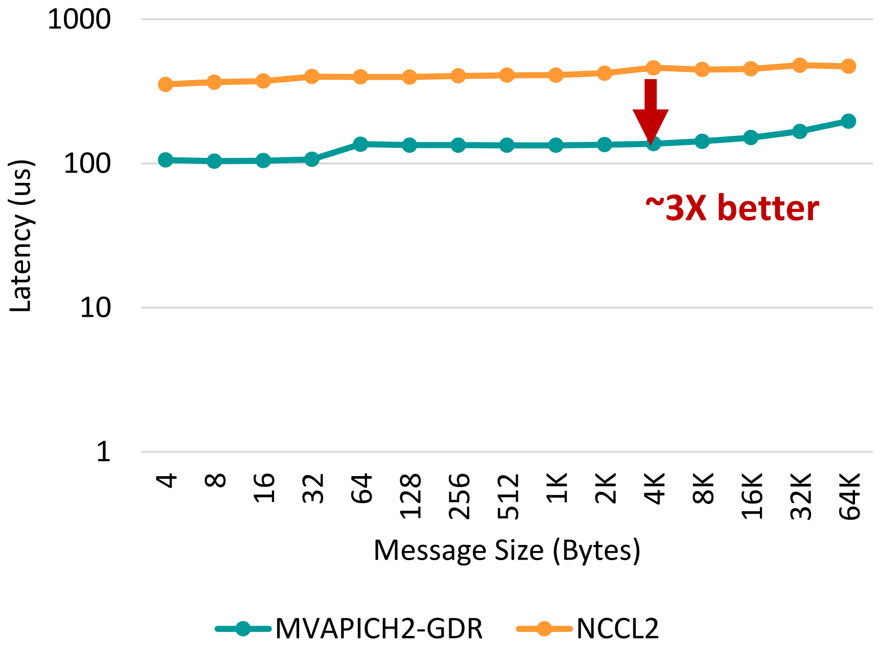



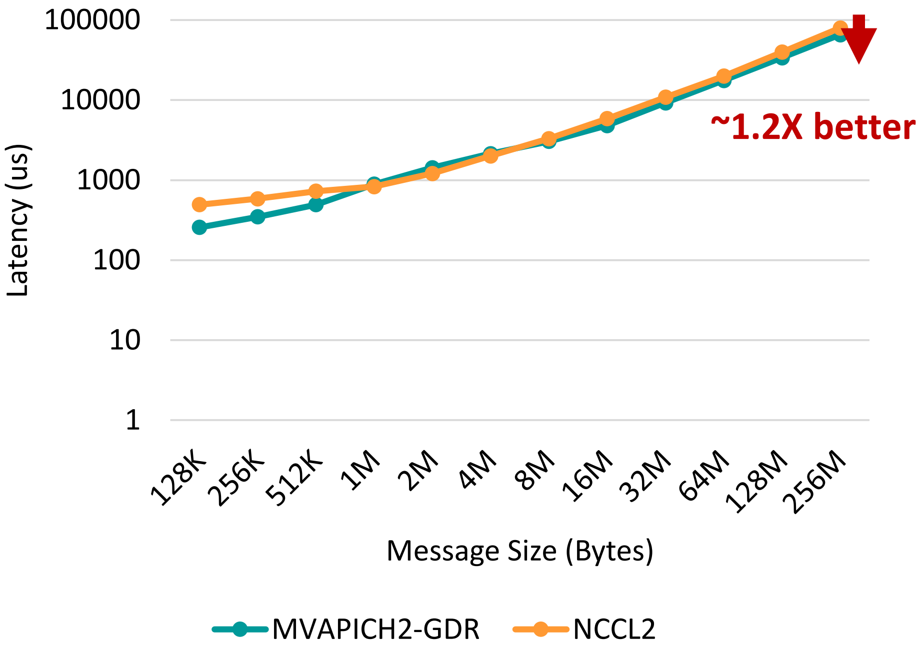




**Available since MVAPICH2-GDR 2.3a*

# 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**

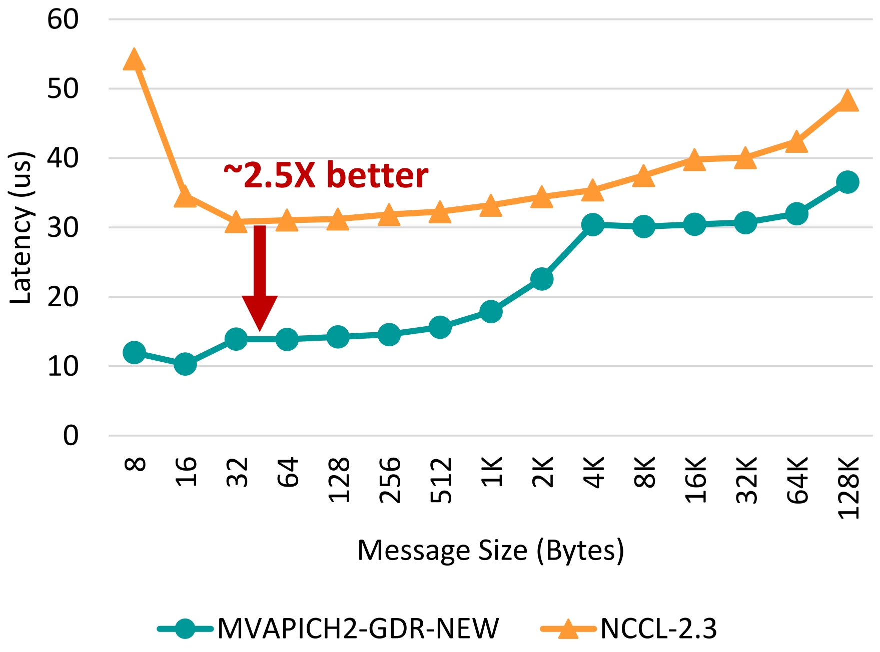


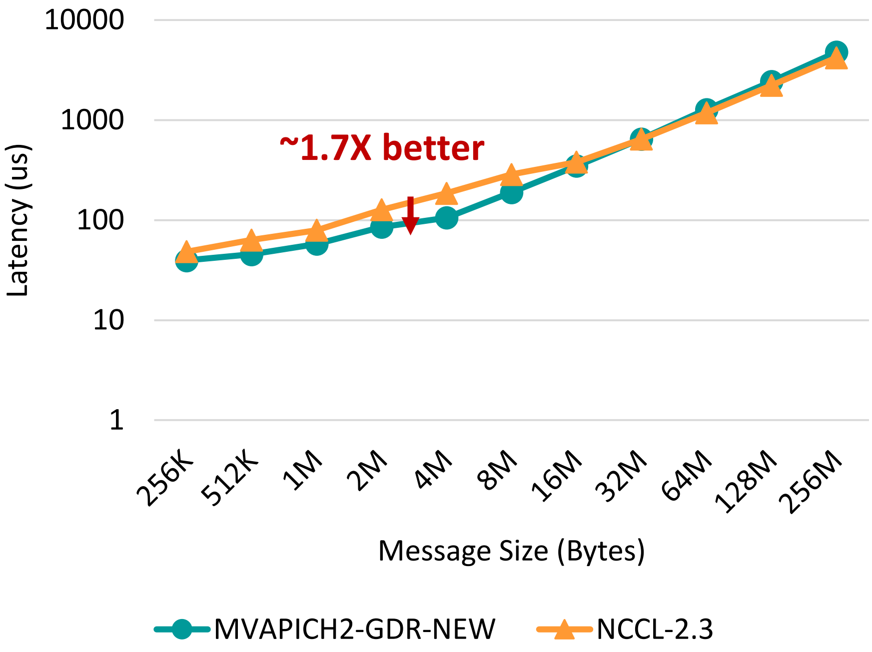




***Platform: Intel Xeon (Broadwell) nodes equipped with a dual-socket CPU, 1 K-80 GPUs, and EDR InfiniBand Inter-connect***

# MVAPICH2-GDR vs. NCCL2 – Allreduce on DGX-2 (Preliminary Results)

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



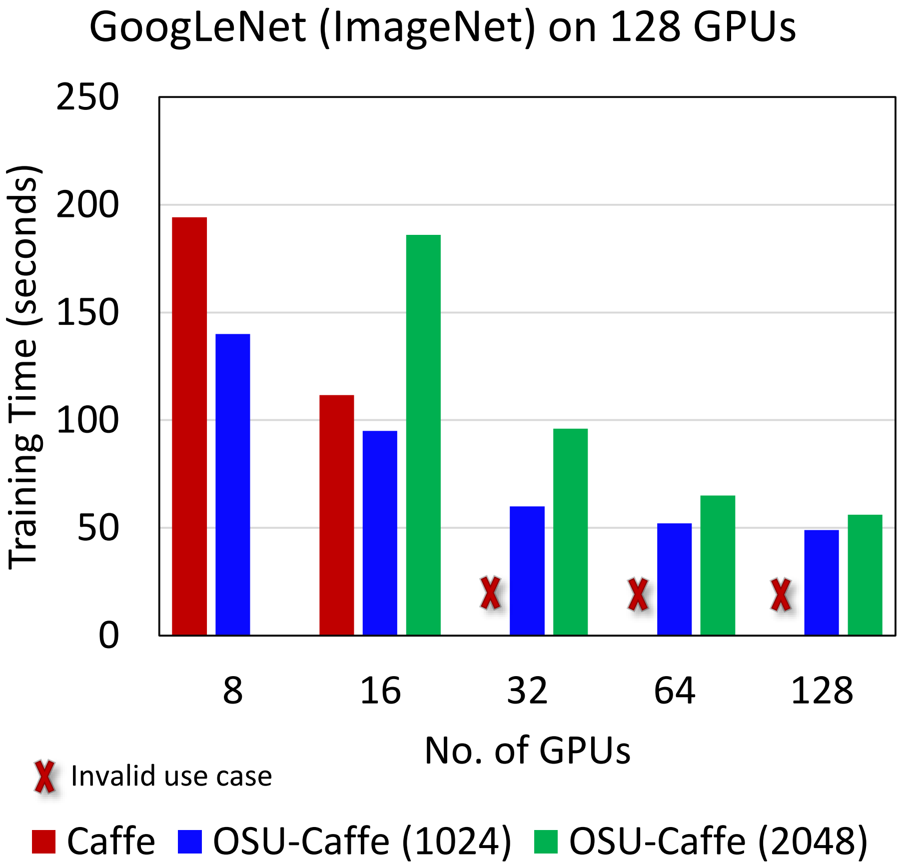

**Platform: Nvidia DGX-2 system (16 Nvidia Volta GPUs connected with NVSwitch), CUDA 9.2**

# OSU-Caffe: Scalable Deep Learning

- Caffe : A flexible and layered Deep Learning framework.
- Benefits and Weaknesses
  - Multi-GPU Training within a single node
  - Performance degradation for GPUs across different sockets
  - Limited Scale-out
- OSU-Caffe: MPI-based Parallel Training
  - Enable Scale-up (within a node) and Scale-out (across multi-GPU nodes)
  - Scale-out on 64 GPUs for training CIFAR-10 network on CIFAR-10 dataset
  - Scale-out on 128 GPUs for training GoogLeNet network on ImageNet dataset

OSU-Caffe publicly available from

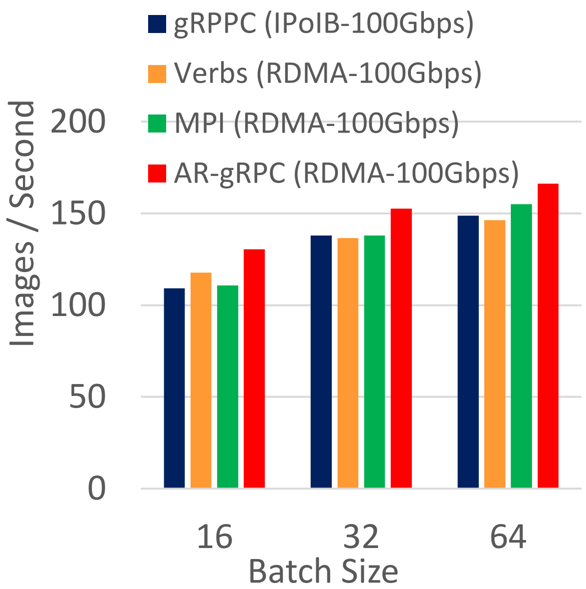
[http://hidl.cse.ohio-state.edu/](http://hidl.cse.ohio-state.edu/)



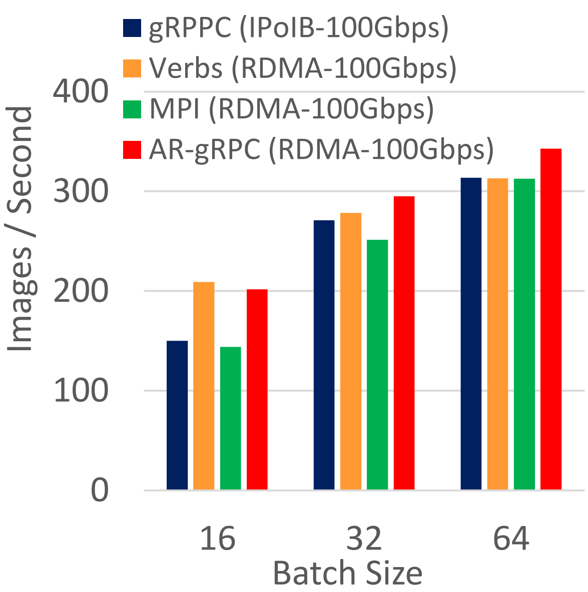
# RDMA-TensorFlow Distribution

- High-Performance Design of TensorFlow over RDMA-enabled Interconnects
  - High performance RDMA-enhanced design with native InfiniBand support at the verbs-level for gRPC and TensorFlow
  - RDMA-based data communication
  - Adaptive communication protocols
  - Dynamic message chunking and accumulation
  - Support for RDMA device selection
  - Easily configurable for different protocols (native InfiniBand and IPoIB)
- Current release: 0.9.1
  - Based on Google TensorFlow 1.3.0
  - Tested with
    - Mellanox InfiniBand adapters (e.g., EDR)
    - NVIDIA GPGPU K80
    - Tested with CUDA 8.0 and CUDNN 5.0
  - http://hidl.cse.ohio-state.edu

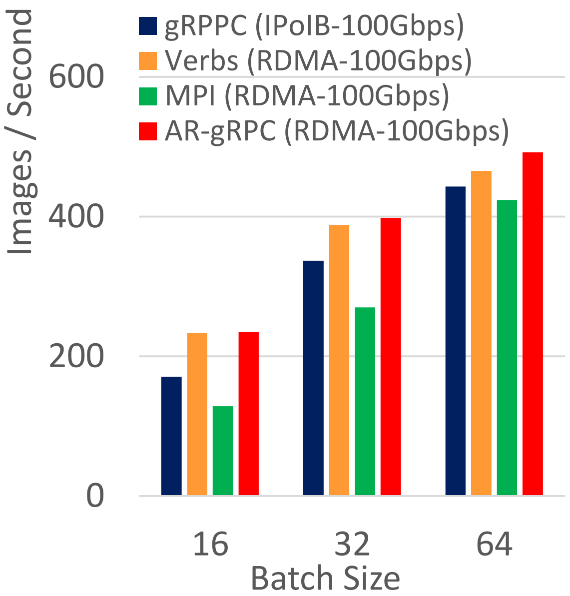
# Performance Benefit for RDMA-TensorFlow (Inception3)




**4 Nodes (8 GPUS)**




**8 Nodes (16 GPUS)**



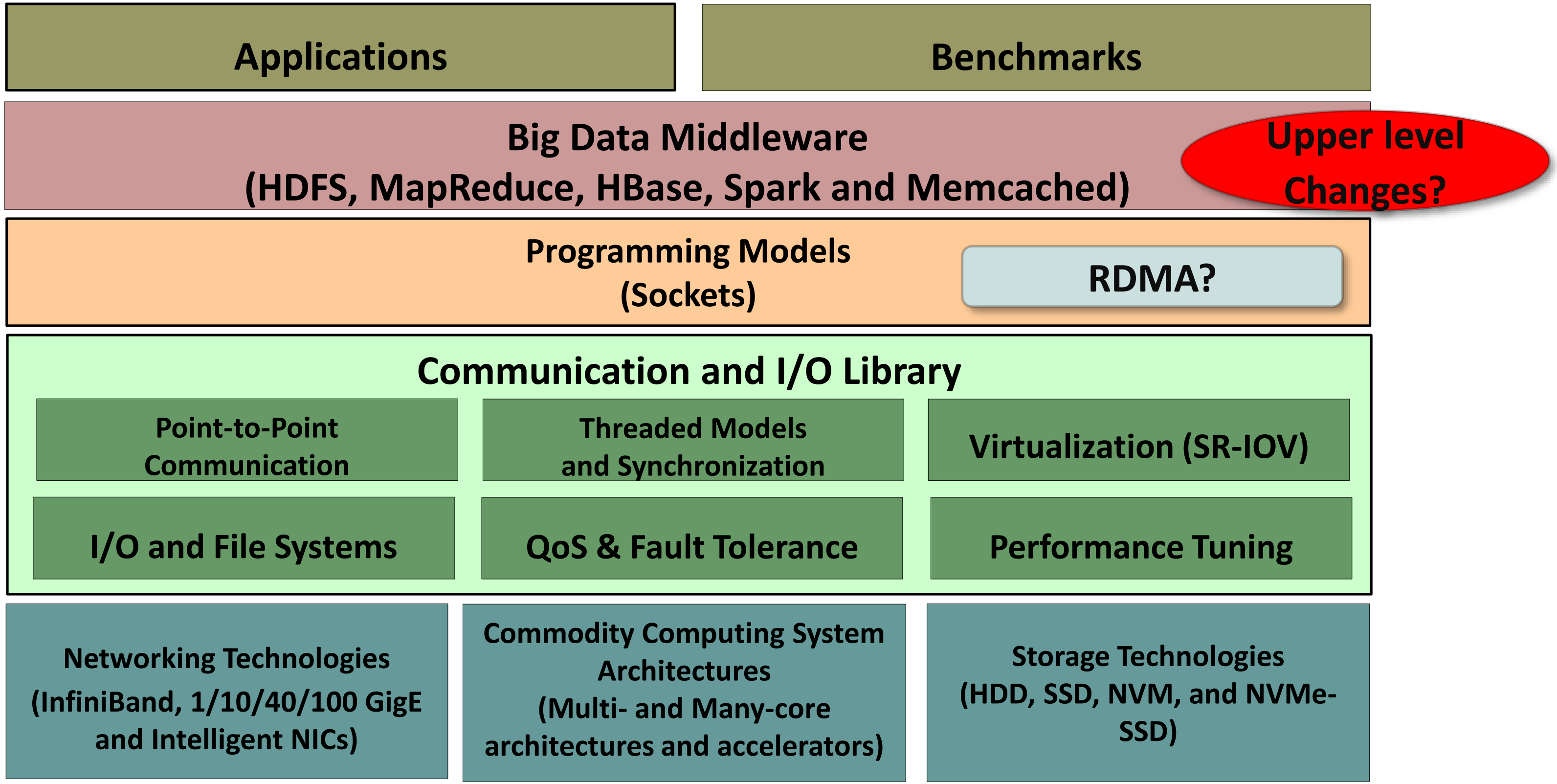

**12 Nodes (24 GPUS)**

- TensorFlow Inception3 performance evaluation on an IB EDR cluster
  - Up to 20% performance speedup over Default gRPC (IPoIB) for 8 GPUs
  - Up to 34% performance speedup over Default gRPC (IPoIB) for 16 GPUs
  - Up to 37% performance speedup over Default gRPC (IPoIB) for 24 GPUs

# HPC, Big Data, Deep Learning, and Cloud

- Traditional HPC
  - Message Passing Interface (MPI), including MPI + OpenMP
  - Exploiting Accelerators
- Deep Learning
  - Caffe, CNTK, TensorFlow, and many more
- Big Data/Enterprise/Commercial Computing
  - Spark and Hadoop (HDFS, HBase, MapReduce)
  - Deep Learning over Big Data (DLoBD)
- Cloud for HPC and BigData
  - Virtualization with SR-IOV and Containers

# Designing Communication and I/O Libraries for Big Data Systems: Challenges



# The High-Performance Big Data (HiBD) Project

- RDMA for Apache Spark
- RDMA for Apache Hadoop 2.x (RDMA-Hadoop-2.x)
  - Plugins for Apache, Hortonworks (HDP) and Cloudera (CDH) Hadoop distributions
- RDMA for Apache HBase
- RDMA for Memcached (RDMA-Memcached)
- RDMA for Apache Hadoop 1.x (RDMA-Hadoop)
- OSU HiBD-Benchmarks (OHB)
  - HDFS, Memcached, HBase, and Spark Micro-benchmarks
- http://hibd.cse.ohio-state.edu
- Users Base: 290 organizations from 34 countries
- More than 28,500 downloads from the project site

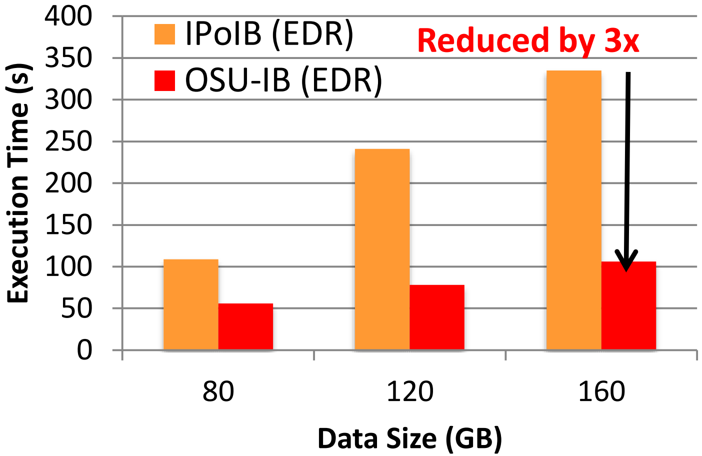
**Available for InfiniBand and RoCE**

**Also run on Ethernet**

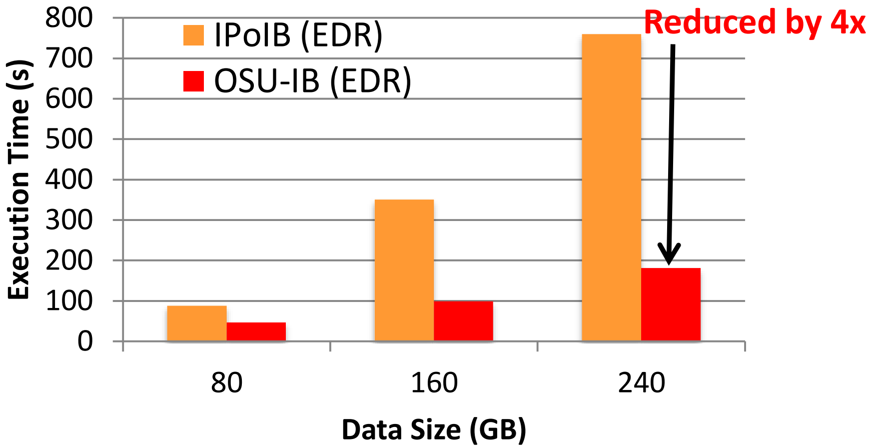
**Available for x86 and OpenPOWER**

**Support for Singularity and Docker**

# Performance Numbers of RDMA for Apache Hadoop 2.x – RandomWriter & TeraGen in OSU-RI2 (EDR)




RandomWriter

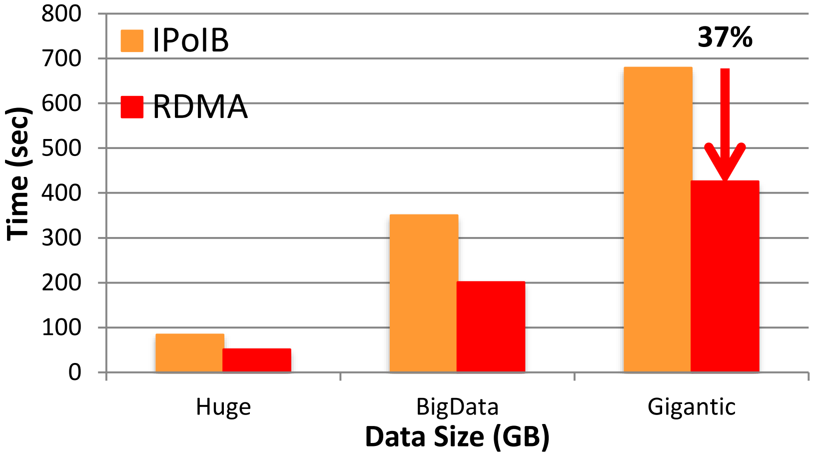



TeraGen

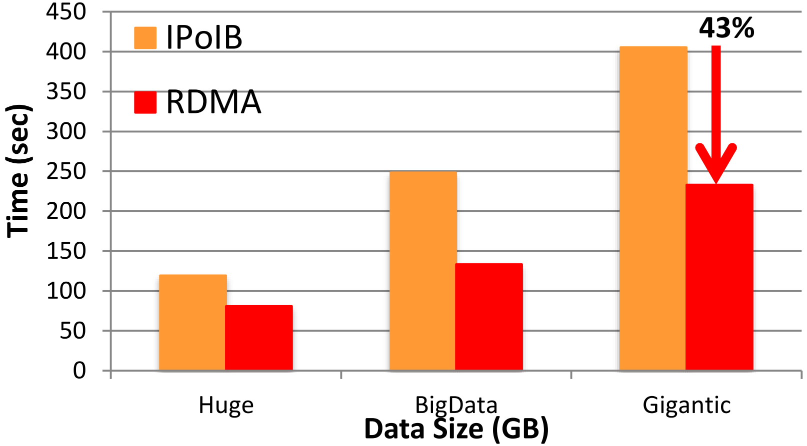
**Cluster with 8 Nodes with a total of 64 maps**

- RandomWriter
  - **3x** improvement over IPoIB for 80-160 GB file size
- TeraGen
  - **4x** improvement over IPoIB for 80-240 GB file size

# Performance Evaluation of RDMA-Spark on SDSC Comet – HiBench PageRank



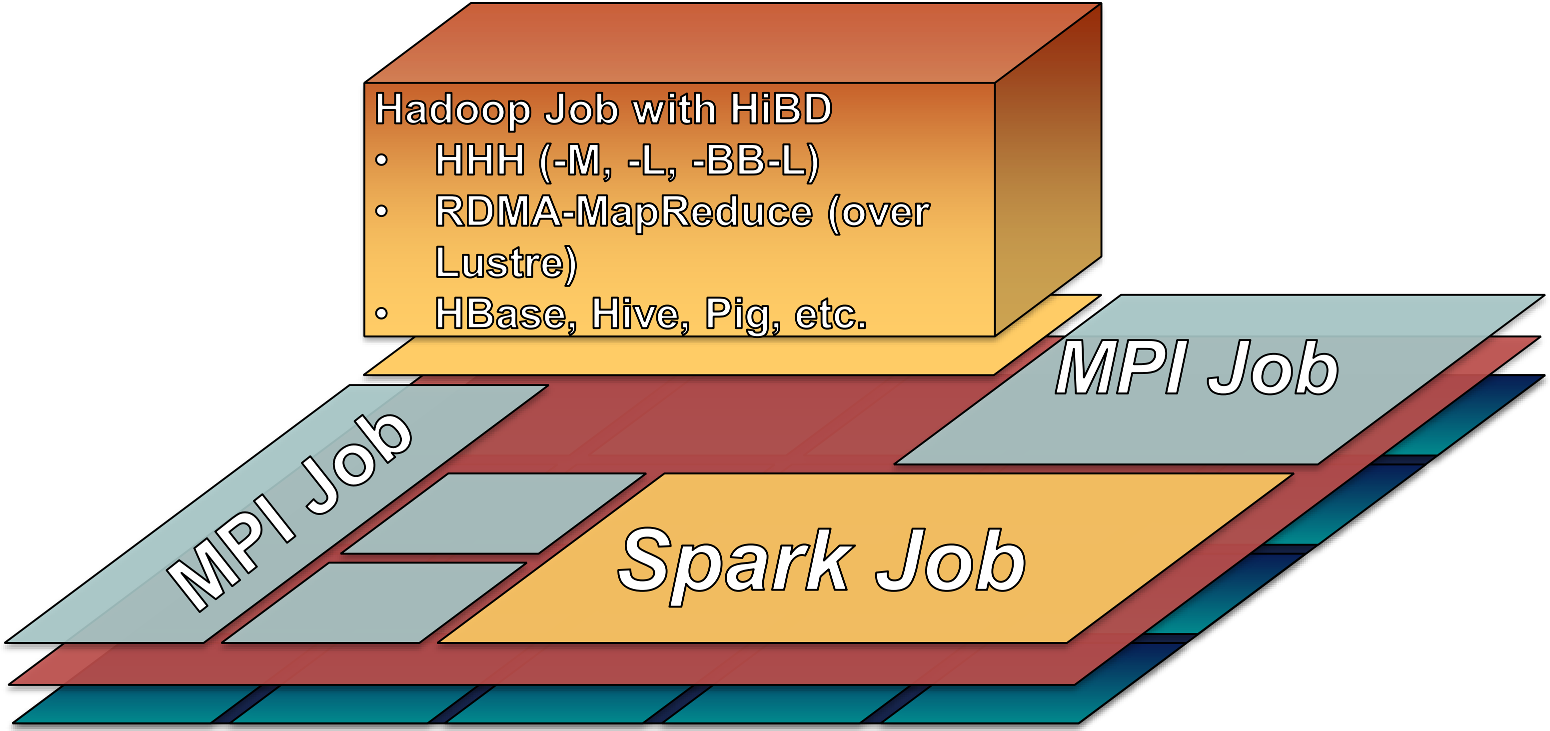

**32 Worker Nodes, 768 cores, PageRank Total Time**



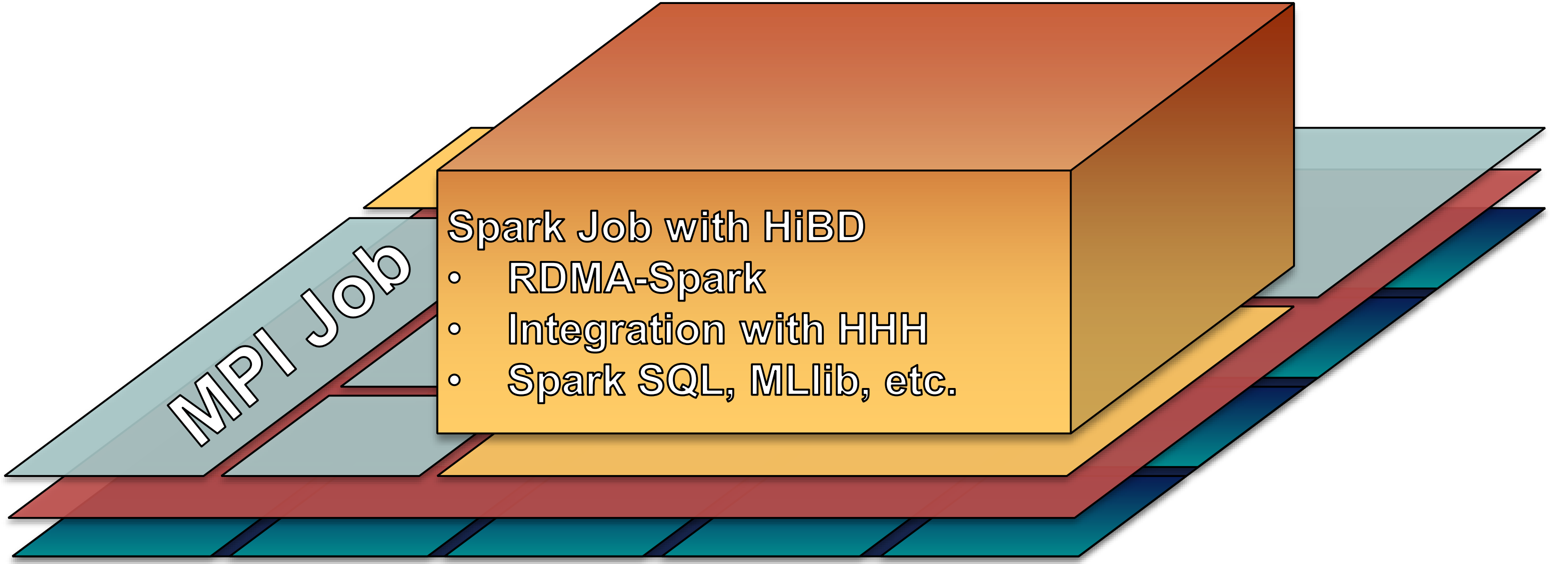

**64 Worker Nodes, 1536 cores, PageRank Total Time**

- InfiniBand FDR, SSD, 32/64 Worker Nodes, 768/1536 Cores, (768/1536M 768/1536R)
- RDMA vs. IPoIB with 768/1536 concurrent tasks, single SSD per node.
  - 32 nodes/768 cores: Total time reduced by 37% over IPoIB (56Gbps)
  - 64 nodes/1536 cores: Total time reduced by 43% over IPoIB (56Gbps)

# Using HiBD Packages on Existing HPC Infrastructure



# Using HiBD Packages on Existing HPC Infrastructure



# Deep Learning over Big Data (DLoBD)

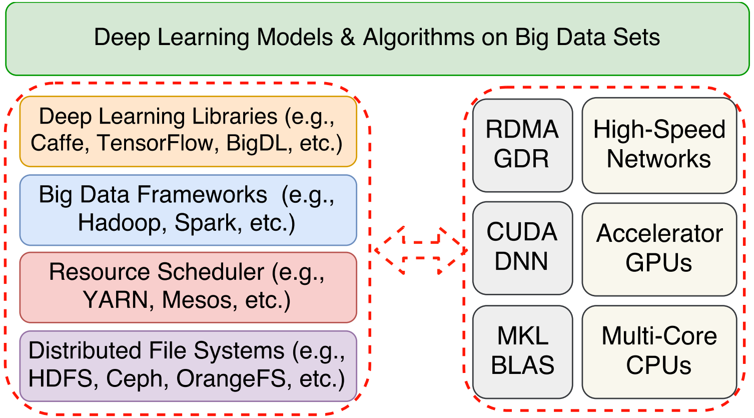
- Deep Learning over Big Data (**DLoBD**) is one of the most efficient analyzing paradigms
- More and more deep learning tools or libraries (e.g., Caffe, TensorFlow) start running over big data stacks, such as Apache Hadoop and Spark
- **Benefits** of the DLoBD approach
  - Easily build a powerful data analytics **pipeline**
    - E.g., Flickr DL/ML Pipeline, "*How Deep Learning Powers Flickr*", http://bit.ly/1KIDfof

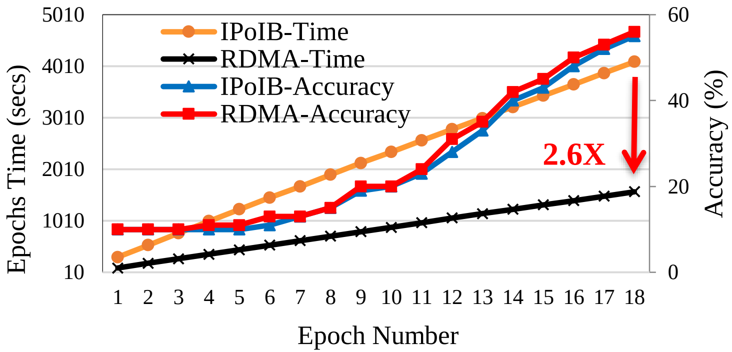
(1) Prepare Datasets @Scale → (2) Deep Learning @Scale → (3) Non-deep learning analytics @Scale → (4) Apply ML model @Scale

  - Better data **locality**
  - Efficient resource sharing and **cost effective**

# High-Performance Deep Learning over Big Data (DLoBD) Stacks

- Benefits of Deep Learning over Big Data (DLoBD)
  - Easily integrate deep learning components into Big Data processing workflow
  - Easily access the stored data in Big Data systems
  - No need to set up new dedicated deep learning clusters; Reuse existing big data analytics clusters
- Challenges
  - Can RDMA-based designs in DLoBD stacks improve performance, scalability, and resource utilization on high-performance interconnects, GPUs, and multi-core CPUs?
  - What are the performance characteristics of representative DLoBD stacks on RDMA networks?
- Characterization on DLoBD Stacks
  - CaffeOnSpark, TensorFlowOnSpark, and BigDL
  - IPoIB vs. RDMA; In-band communication vs. Out-of-band communication; CPU vs. GPU; etc.
  - Performance, accuracy, scalability, and resource utilization
  - RDMA-based DLoBD stacks (e.g., BigDL over RDMA-Spark) can achieve 2.6x speedup compared to the IPoIB based scheme, while maintain similar accuracy




X. Lu, H. Shi, M. H. Javed, R. Biswas, and D. K. Panda, Characterizing Deep Learning over Big Data (DLoBD) Stacks on RDMA-capable Networks, HotI 2017.

# HPC, Big Data, Deep Learning, and Cloud

- Traditional HPC
  - Message Passing Interface (MPI), including MPI + OpenMP
  - Exploiting Accelerators
- Deep Learning
  - Caffe, CNTK, TensorFlow, and many more
- Big Data/Enterprise/Commercial Computing
  - Spark and Hadoop (HDFS, HBase, MapReduce)
  - Deep Learning over Big Data (DLoBD)
- Cloud for HPC and BigData
  - Virtualization with SR-IOV and Containers

# Can HPC and Virtualization be Combined?

- Virtualization has many benefits
  - Fault-tolerance
  - Job migration
  - Compaction
- Have not been very popular in HPC due to overhead associated with Virtualization
- New SR-IOV (Single Root – IO Virtualization) support available with Mellanox InfiniBand adapters changes the field
- Enhanced MVAPICH2 support for SR-IOV
- MVAPICH2-Virt 2.2 supports:
  - OpenStack, Docker, and singularity

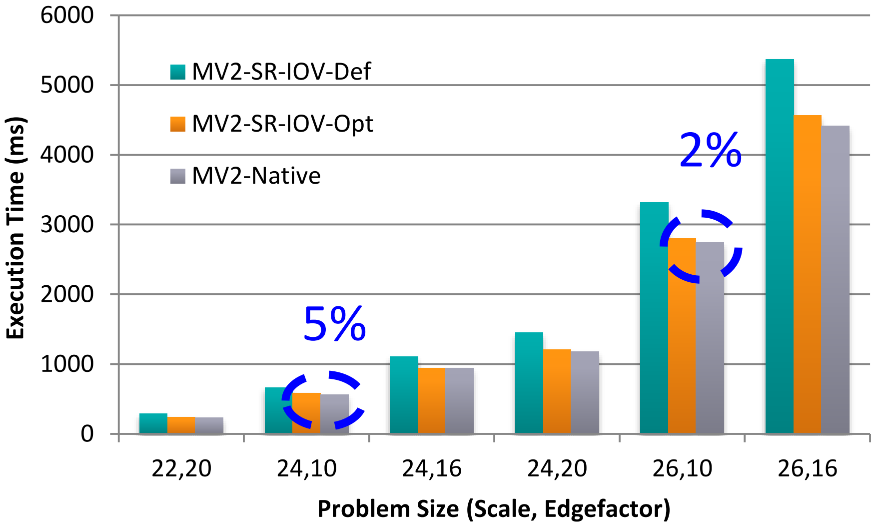

**J. Zhang, X. Lu, J. Jose, R. Shi and D. K. Panda, Can Inter-VM Shmem Benefit MPI Applications on SR-IOV based Virtualized InfiniBand Clusters? EuroPar'14**

**J. Zhang, X. Lu, J. Jose, M. Li, R. Shi and D.K. Panda, High Performance MPI Library over SR-IOV enabled InfiniBand Clusters, HiPC'14**

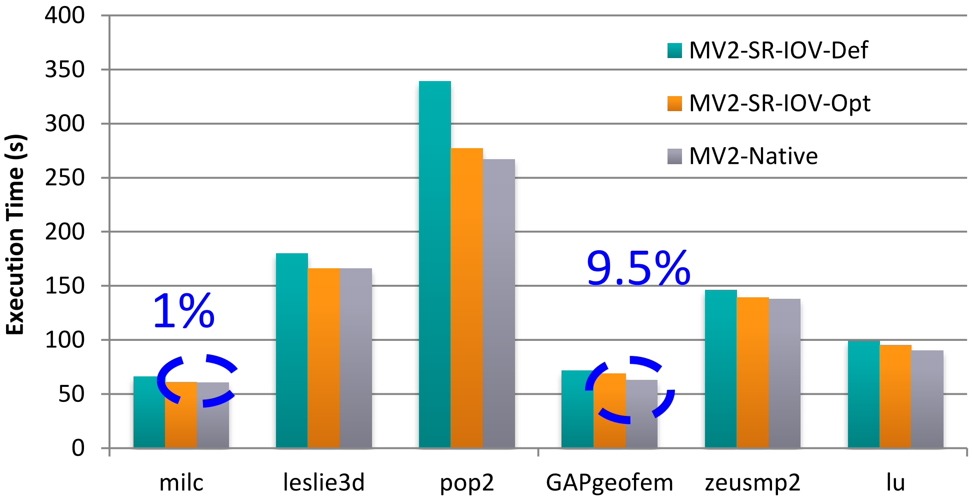
**J. Zhang, X .Lu, M. Arnold and D. K. Panda, MVAPICH2 Over OpenStack with SR-IOV: an Efficient Approach to build HPC Clouds, CCGrid'15**

# Application-Level Performance on Chameleon

**A Release for Azure Coming Soon**



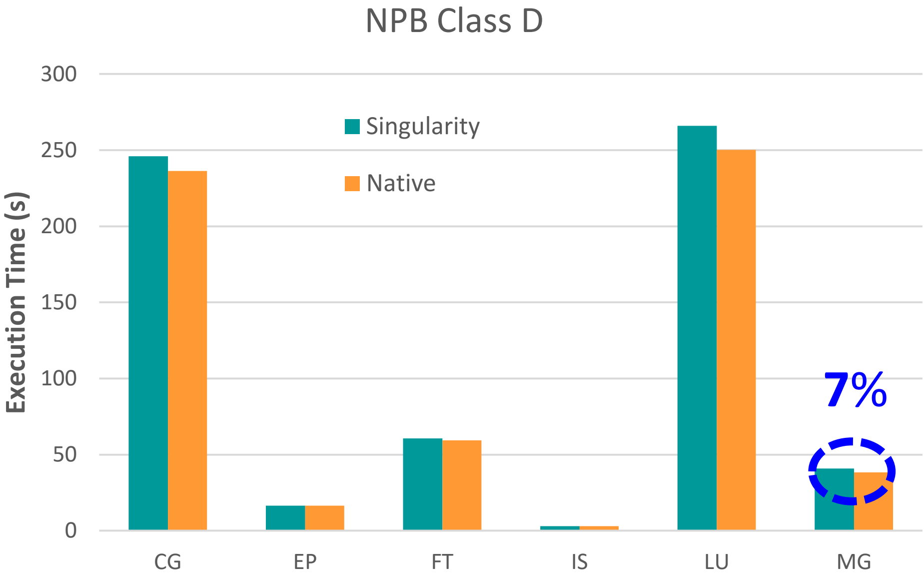

Graph500

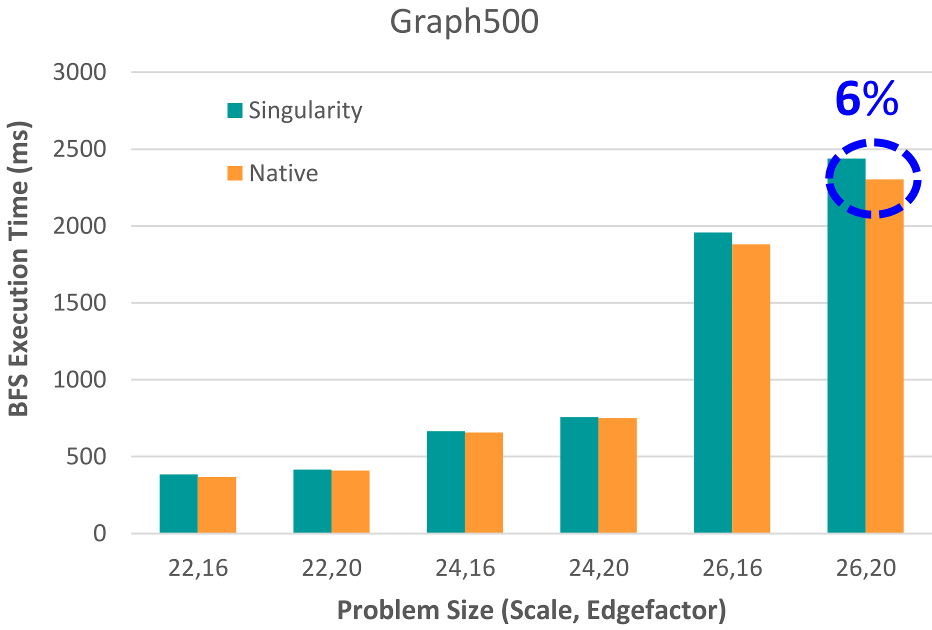



SPEC MPI2007

- 32 VMs, 6 Core/VM
- Compared to Native, 2-5% overhead for Graph500 with 128 Procs
- Compared to Native, 1-9.5% overhead for SPEC MPI2007 with 128 Procs

# Application-Level Performance on Singularity with MVAPICH2

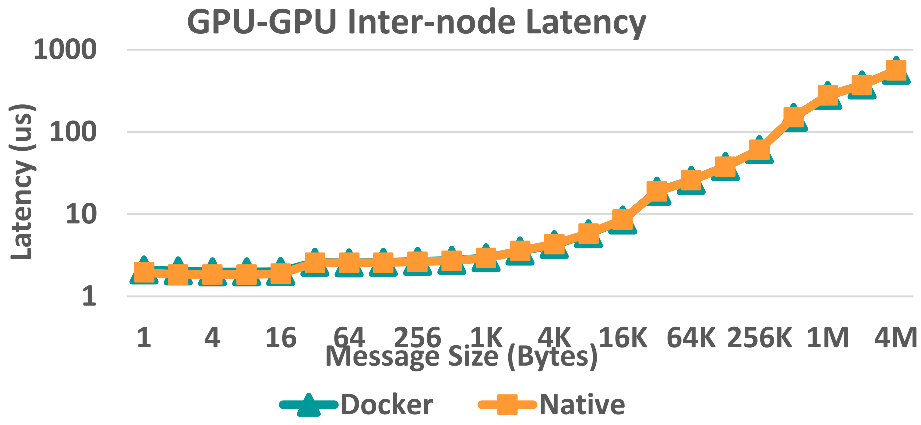


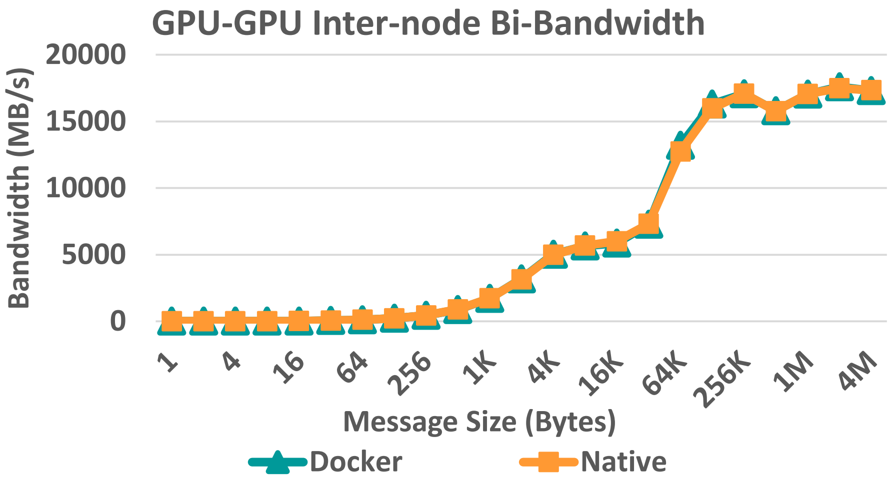



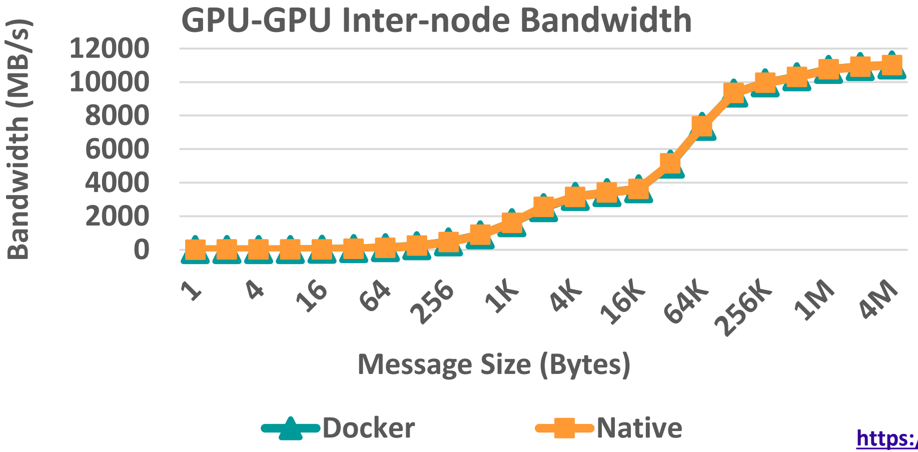

- 512 Processes across 32 nodes
- Less than 7% and 6% overhead for NPB and Graph500, respectively

**J. Zhang, X .Lu and D. K. Panda, Is Singularity-based Container Technology Ready for Running MPI Applications on HPC Clouds?, UCC '17, Best Student Paper Award**

# MVAPICH2-GDR on Container with Negligible Overhead




**MVAPICH2-GDR-2.3a**
**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**

Works with NVIDIA HPC Container Maker
https://github.com/NVIDIA/hpc-container-maker/blob/master/recipes/hpcbase-pgi-mvapich2.py

# 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) this year

# Multiple Positions Available in My Group

- Looking for Bright and Enthusiastic Personnel to join as
  - Post-Doctoral Researchers
  - PhD Students
  - MPI Programmer/Software Engineer
  - Deep Learning/Big Data Programmer/Software Engineer
- If interested, please contact me at this conference and/or send an e-mail to panda@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.)
- S. Chakraborthy  (Ph.D.)
- C.-H. Chu (Ph.D.)
- S. Guganani (Ph.D.)
- J. Hashmi (Ph.D.)
- H. Javed (Ph.D.)
- P. Kousha (Ph.D.)
- D. Shankar (Ph.D.)
- H. Shi (Ph.D.)

***Current Students (Undergraduate)***

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

***Current Research Scientists***

- X. Lu
- H. Subramoni

***Current Post-doc***

- 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

***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/