MVAPICH

MPI, PGAS and Hybrid MPI+PGAS Library

High-Performance MPI and Deep Learning on OpenPOWER Platform

Talk at OpenPOWER SUMMIT (Mar '18)

by

Dhabaleswar K. (DK) Panda

The Ohio State University

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

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

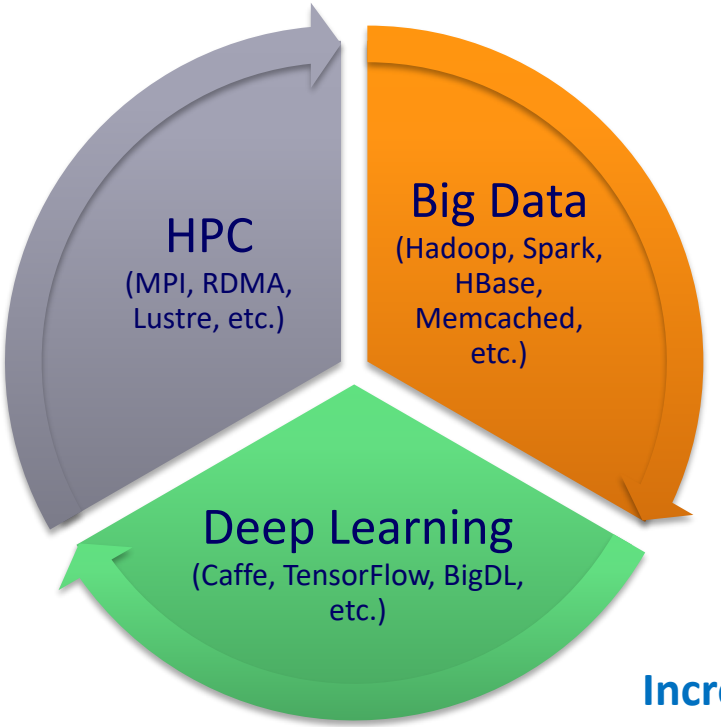
Hari Subramoni

The Ohio State University

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

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

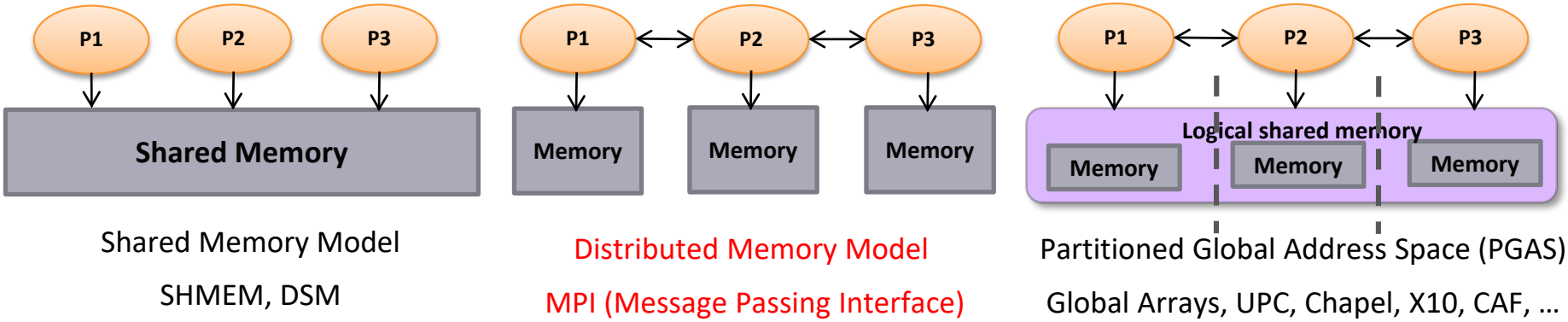
Increasing Usage of HPC, Big Data and Deep Learning



Convergence of HPC, Big Data, and Deep Learning!

Increasing Need to Run these applications on the Cloud!!

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, …

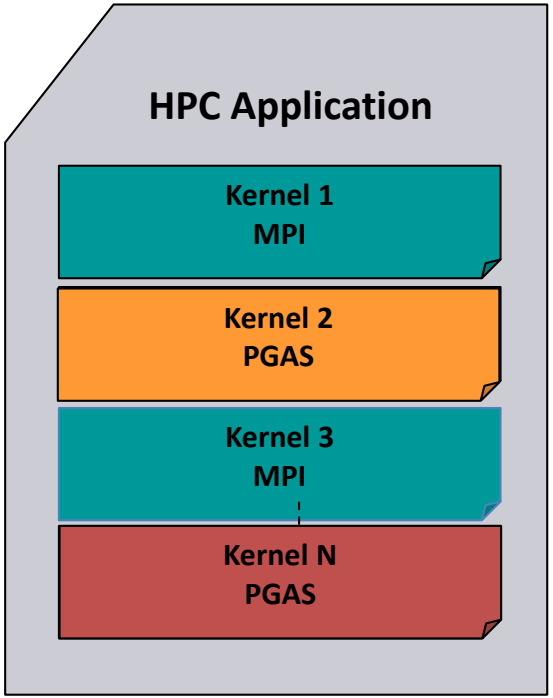
- Programming models provide abstract machine models
- Models can be mapped on different types of systems
  - e.g. Distributed Shared Memory (DSM), MPI within a node, etc.
- PGAS models and Hybrid MPI+PGAS models are gradually receiving importance

Partitioned Global Address Space (PGAS) Models

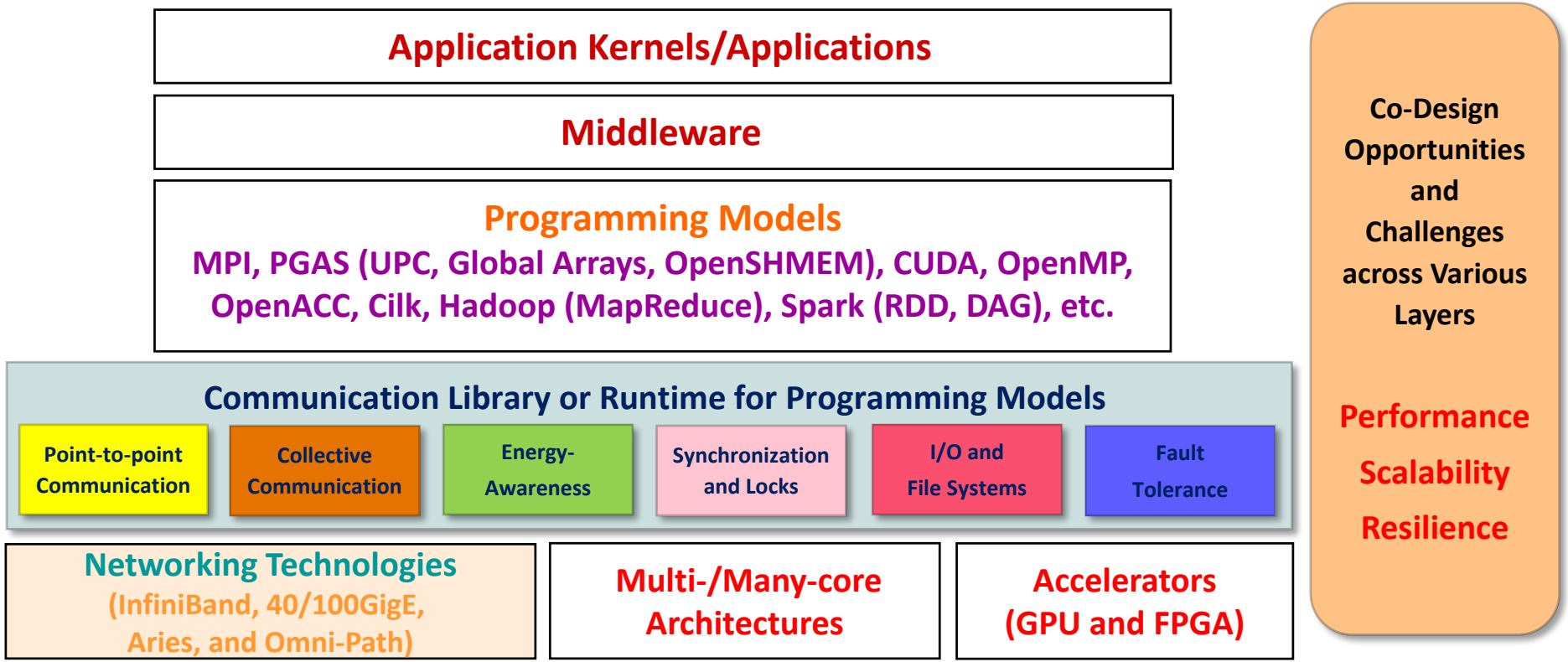
- Key features
  - Simple shared memory abstractions
  - Light weight one-sided communication
  - Easier to express irregular communication
- Different approaches to PGAS
  - Languages
    - Unified Parallel C (UPC)
    - Co-Array Fortran (CAF)
    - X10
    - Chapel
  - Libraries
    - OpenSHMEM
    - UPC++
    - Global Arrays

Hybrid (MPI+PGAS) Programming

- Application sub-kernels can be re-written in MPI/PGAS based on communication characteristics
- Benefits:
  - Best of Distributed Computing Model
  - Best of Shared Memory Computing Model



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



MPI, PGAS and Deep Learning Support for OpenPOWER

- Message Passing Interface (MPI) Support
- Support for PGAS and MPI + PGAS (OpenSHMEM, UPC)
- Exploiting Accelerators (NVIDIA GPGPUs)
- Support for Deep Learning

Support for Big Data (Hadoop and Spark) on OpenPOWER

(Talk at 2:15 pm)

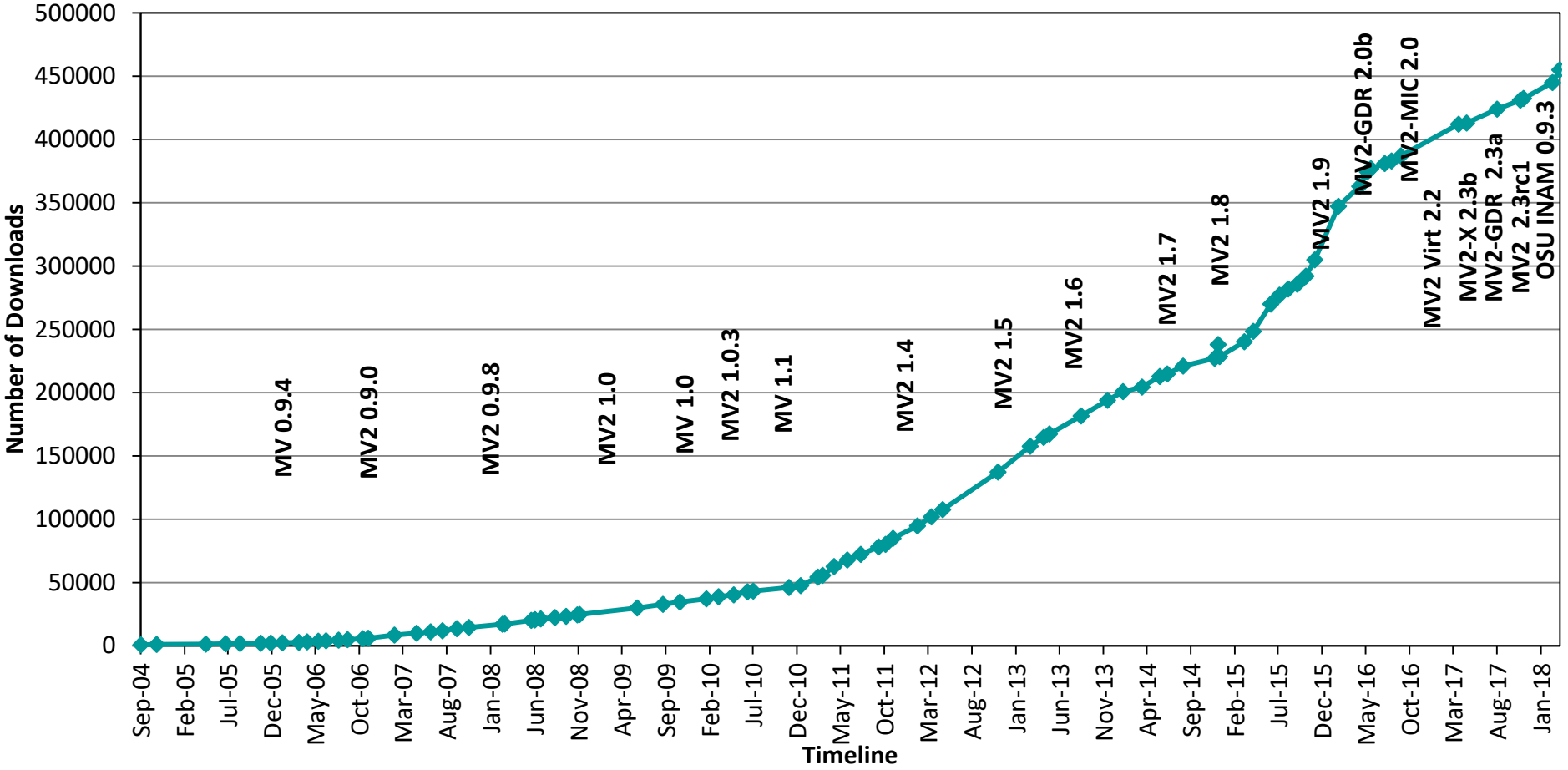
Overview of the MVAPICH2 Project

- 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 2,875 organizations in 85 countries**
  - **More than 455,000 (> 0.45 million) downloads from the OSU site directl**
  - Empowering many TOP500 clusters (Nov '17 ranking)
    - **1st, 10,649,600-core (Sunway TaihuLight) at National Supercomputing Center in Wuxi, China**
    - 9th, 556,104 cores (Oakforest-PACS) in Japan
    - 12th, 368,928-core (Stampede2) at TACC
    - 17th, 241,108-core (Pleiades) at NASA
    - 48th, 76,032-core (Tsubame 2.5) at Tokyo Institute of Technology
  - Available with software stacks of many vendors, Linux Distros (RedHat and SuSE), and OpenHPC
  - **http://mvapich.cse.ohio-state.edu**
- Empowering Top500 systems for over a decade

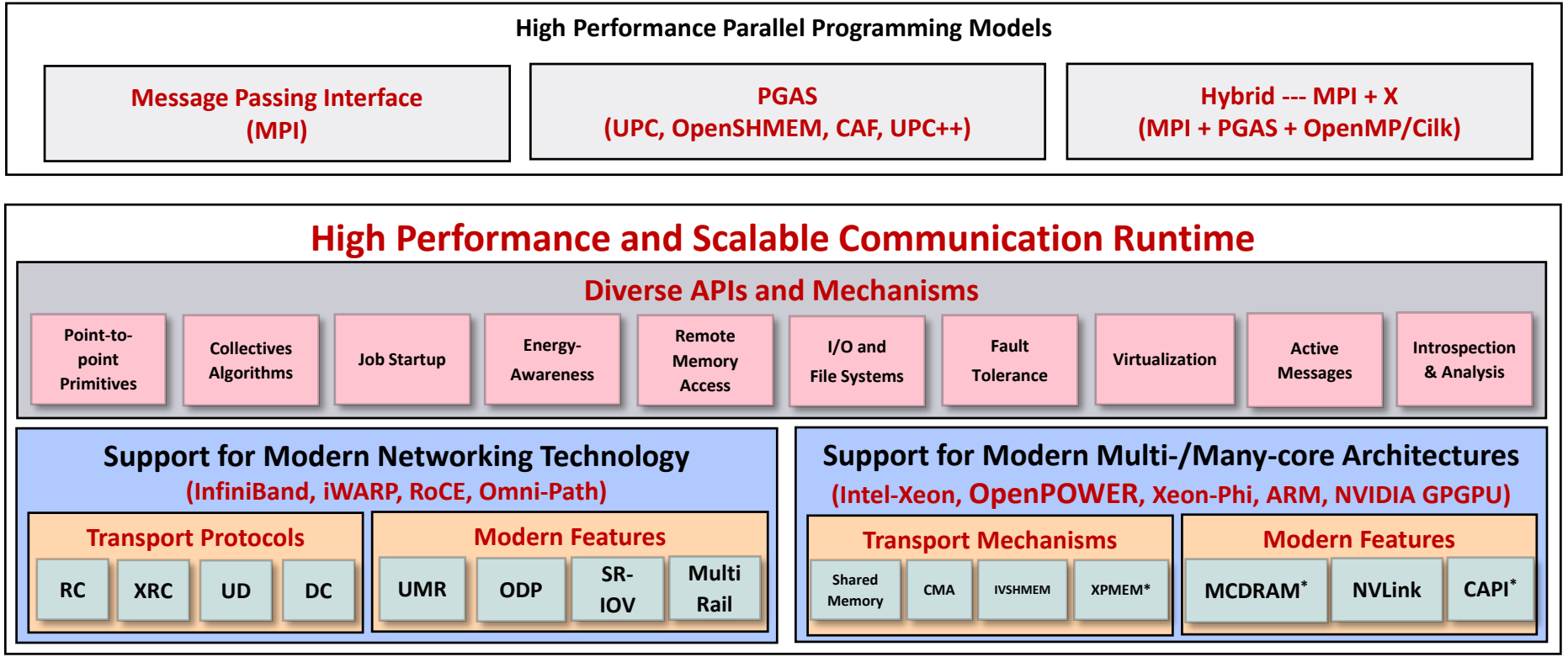
MVAPICH2 Software Family

High-Performance Parallel Programming Libraries	
MVAPICH2	Support for InfiniBand, Omni-Path, Ethernet/iWARP, and RoCE
MVAPICH2-X	Advanced MPI features, OSU INAM, PGAS (OpenSHMEM, UPC, UPC++, and CAF), and MPI+PGAS programming models with unified communication runtime
MVAPICH2-GDR	Optimized MPI for clusters with NVIDIA GPUs
MVAPICH2-Virt	High-performance and scalable MPI for hypervisor and container based HPC cloud
MVAPICH2-EA	Energy aware and High-performance MPI
MVAPICH2-MIC	Optimized MPI for clusters with Intel KNC (being updated with KNL-based designs)
Microbenchmarks	
OMB	Microbenchmarks suite to evaluate MPI and PGAS (OpenSHMEM, UPC, and UPC++) libraries for CPUs and GPUs
Tools	
OSU INAM	Network monitoring, profiling, and analysis for clusters with MPI and scheduler integration
OEMT	Utility to measure the energy consumption of MPI applications

MVAPICH2 Release Timeline and Downloads



Architecture of MVAPICH2 Software Family



* Upcoming

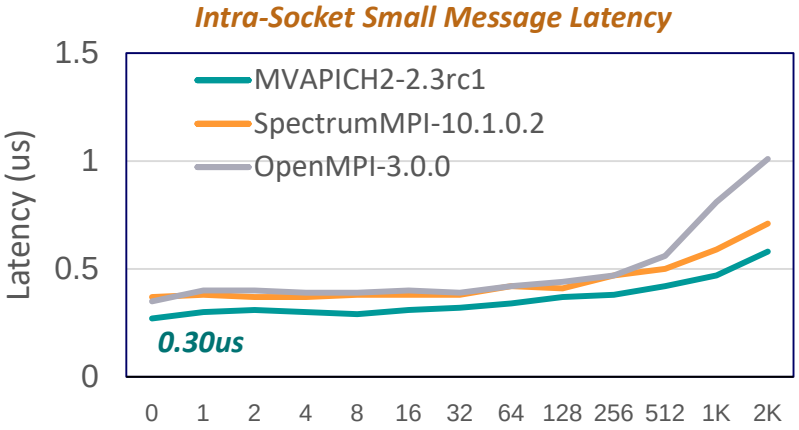
OpenPOWER Platform Support in MVAPICH2 Libraries

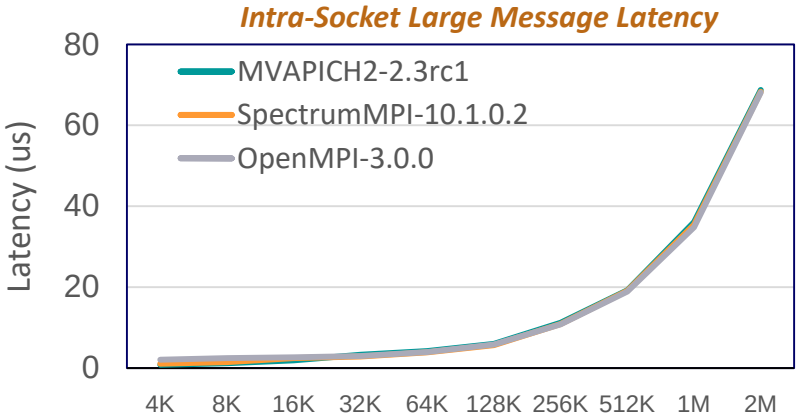
- MVAPICH2
  - Basic MPI support
    - since MVAPICH2 2.2rc1 (March 2016)
- MVAPICH2-X
  - PGAS (OpenSHMEM and UPC) and Hybrid MPI+PGAS support
    - since MVAPICH2-X 2.2b (March 2016)
  - Advanced Collective Support with CMA
    - Since MVAPICH2-X 2.3b (Oct 2017)
- MVAPICH2-GDR
  - NVIDIA GPGPU support with NVLink (CORAL like systems)
    - Since MVAPICH2-GDR 2.3a (Nov 2017)

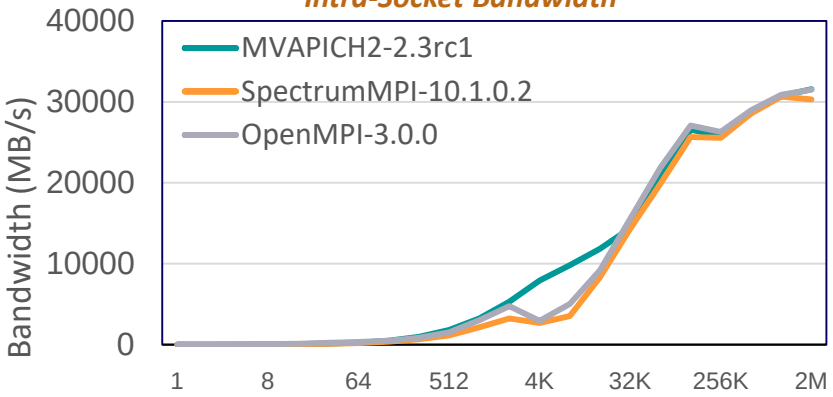
MPI, PGAS and Deep Learning Support for OpenPOWER

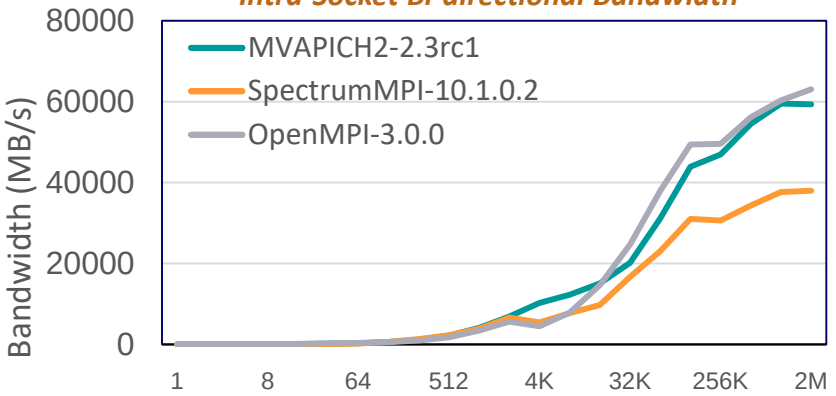
- Message Passing Interface (MPI) Support
  - Point-to-point Inter-node and Intra-node
  - CMA-based Collectives
  - Upcoming XPMEM Support
- Support for PGAS and MPI + PGAS (OpenSHMEM, UPC)
- Exploiting Accelerators (NVIDIA GPGPUs)
- Support for Deep Learning

Intra-node Point-to-Point Performance on OpenPower



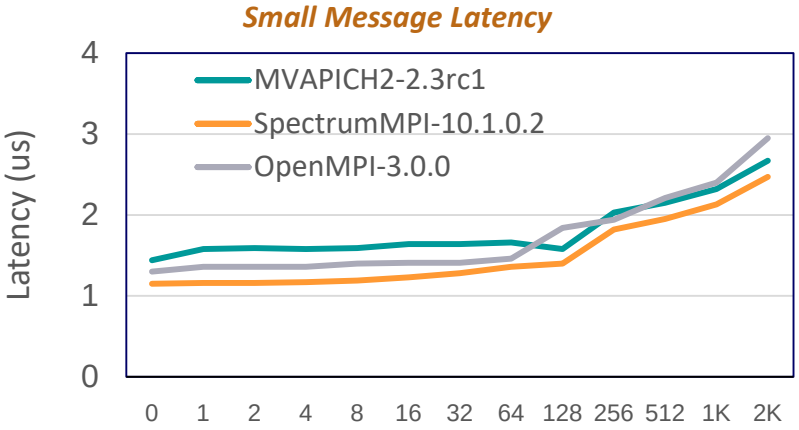


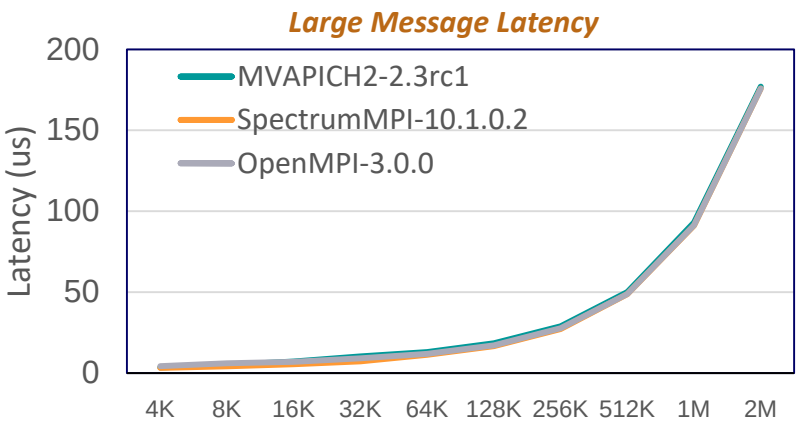


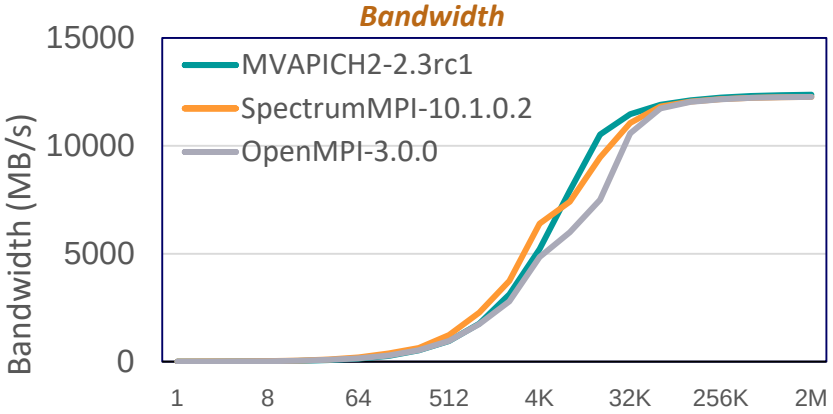


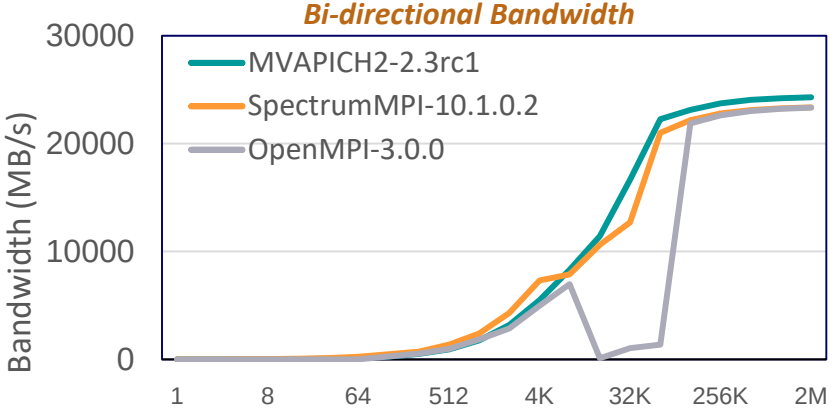
Platform: Two nodes of OpenPOWER (Power8-ppc64le) CPU using Mellanox EDR (MT4115) HCA

Inter-node Point-to-Point Performance on OpenPower



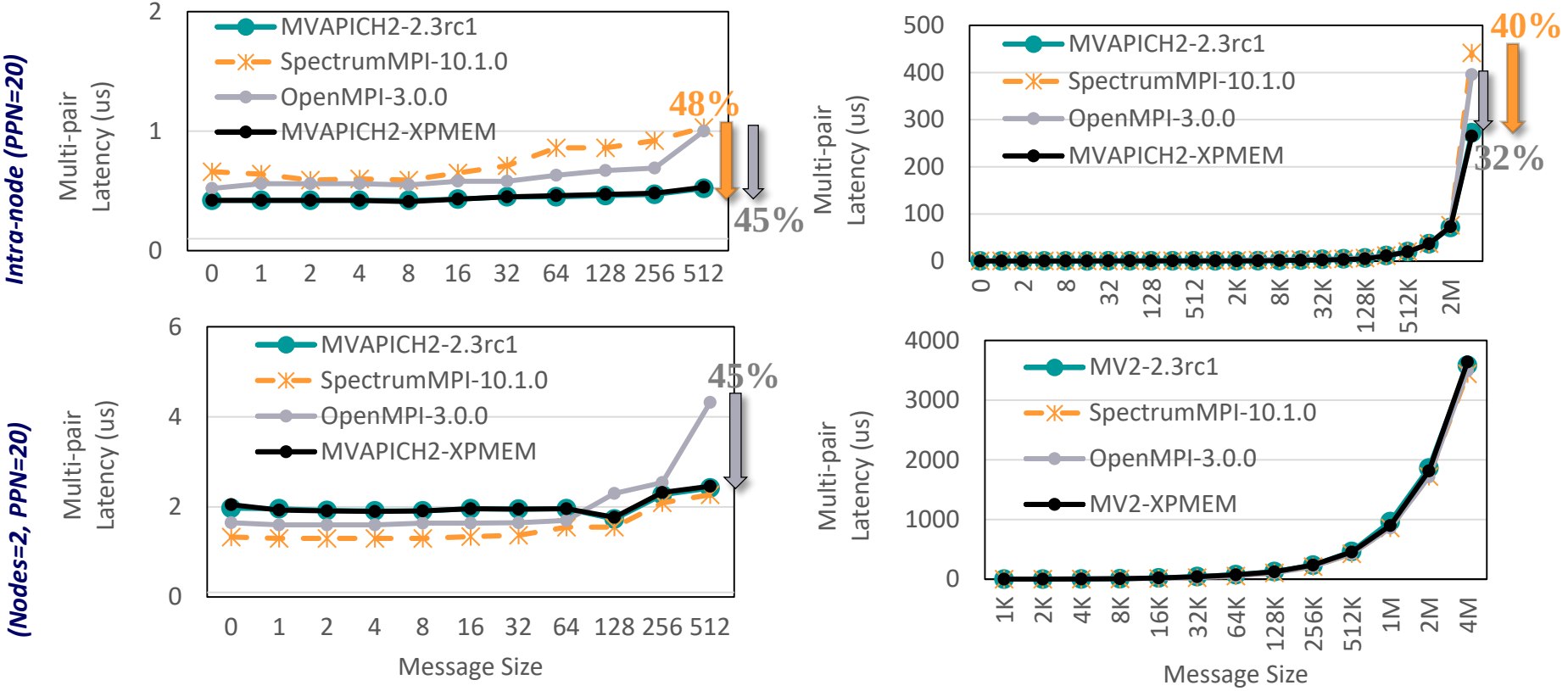






Platform: Two nodes of OpenPOWER (Power8-ppc64le) CPU using Mellanox EDR (MT4115) HCA

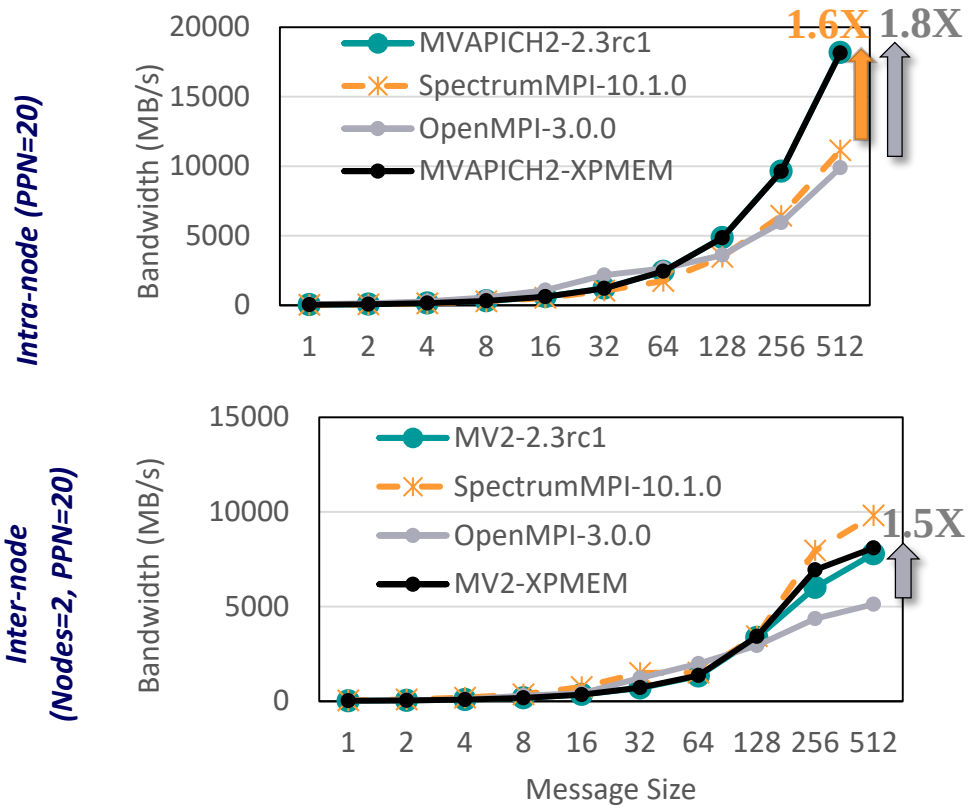
Multi-Pair Latency Comparison



- **Basic point-to-point multi-pair latency comparison**
  - ***Up to 48% and 45%*** performance improvement over Spectrum MPI and OpenMPI for intra-/inter-node

Optimized Runtime Parameters: MV2_CPU_BINDING_POLICY=hybrid MV2_HYBRID_BINDING_POLICY=bunch

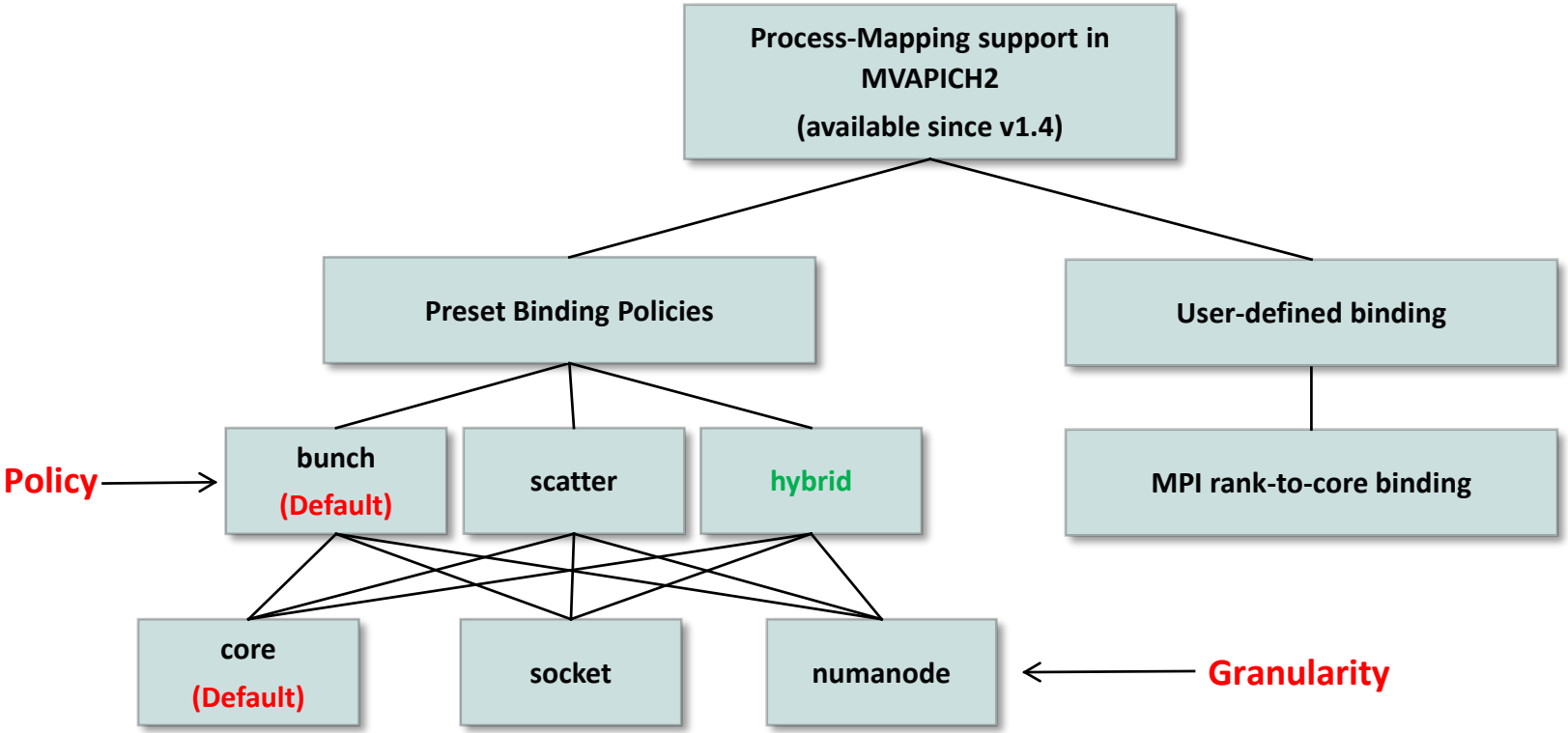
Multi-pair Bandwidth Comparison



- **Basic point-to-point multi-pair bandwidth comparison**
  - ***Up to 83% and 62%*** performance improvement over Spectrum MPI and OpenMPI for intra-/inter-node

Optimized Runtime Parameters: MV2_CPU_BINDING_POLICY=hybrid MV2_HYBRID_BINDING_POLICY=bunch

Process Mapping Support in MVAPICH2 for Multi-Core Platforms



- MVAPICH2 detects processor architecture at job-launch

Process and Thread binding policies in Hybrid MPI+Threads

- A new process binding policy – "hybrid"
  - MV2_CPU_BINDING_POLICY = hybrid
- A new environment variable for co-locating Threads with MPI Processes
  - MV2_THREADS_PER_PROCESS = *k*
  - Automatically set to OMP_NUM_THREADS if OpenMP is being used
  - Provides a hint to the MPI runtime to spare resources for application threads.
- New variable for threads bindings with respect to parent process and architecture
  - MV2_THREADS_BINDING_POLICY = {linear | compact | spread}
    - Linear – binds MPI ranks and OpenMP threads sequentially (one after the other)
      - Recommended to be used on non-hyperthreaded systems with MPI+OpenMP
    - Compact – binds MPI rank to physical-core and locates respective OpenMP threads on hardware threads
      - Recommended to be used on multi-/many-cores e.g., KNL, POWER8, and hyper-threaded Xeon

User-Defined Process Mapping

- User has complete-control over process-mapping
- To run 4 processes on cores 0, 1, 4, 5:
  - $ mpirun_rsh -np 4 -hostfile hosts MV2_CPU_MAPPING=0:1:4:5 ./a.out
- Use ',' or '-' to bind to a set of cores:
  - $mpirun_rsh -np 64 -hostfile hosts MV2_CPU_MAPPING=0,2-4:1:5:6 ./a.out
- Is process binding working as expected?
  - **MV2_SHOW_CPU_BINDING=1**
    - Display CPU binding information
    - Launcher independent
    - Example
      - MV2_SHOW_CPU_BINDING=1 MV2_CPU_BINDING_POLICY=scatter

        ----------CPU AFFINITY----------

        RANK:0 CPU_SET: 0

        RANK:1 CPU_SET: 8

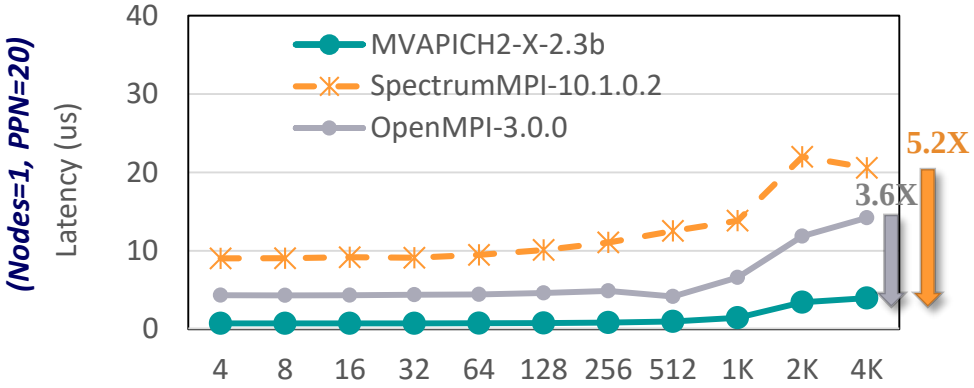
- **Refer to Running with Efficient CPU (Core) Mapping section of MVAPICH2 user guide for more information**
- **http://mvapich.cse.ohio-state.edu/static/media/mvapich/mvapich2-2.3rc1-userguide.html#x1-600006.5**

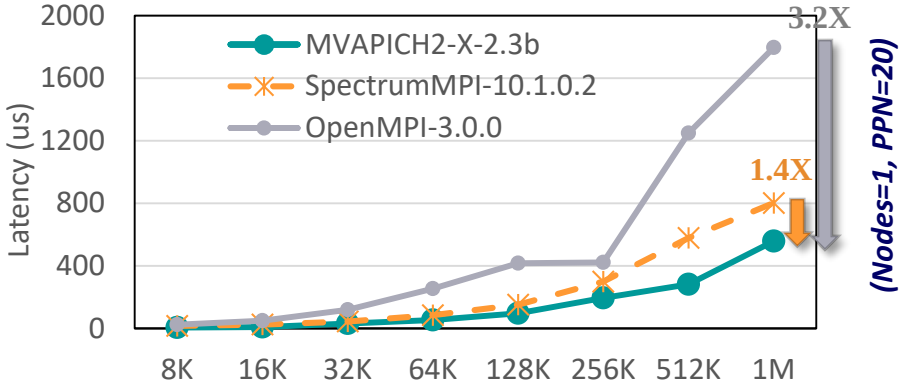
MPI, PGAS and Deep Learning Support for OpenPOWER

- Message Passing Interface (MPI) Support
  - Point-to-point Inter-node and Intra-node
  - CMA-based Collectives
  - Upcoming XPMEM Support
- Support for PGAS and MPI + PGAS (OpenSHMEM, UPC)
- Exploiting Accelerators (NVIDIA GPGPUs)
- Support for Deep Learning

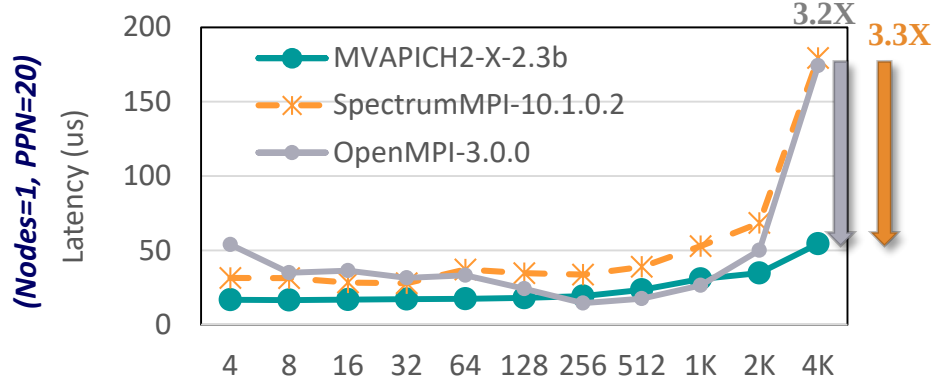
Scalable Collectives with CMA (Intra-node Reduce & AlltoAll)

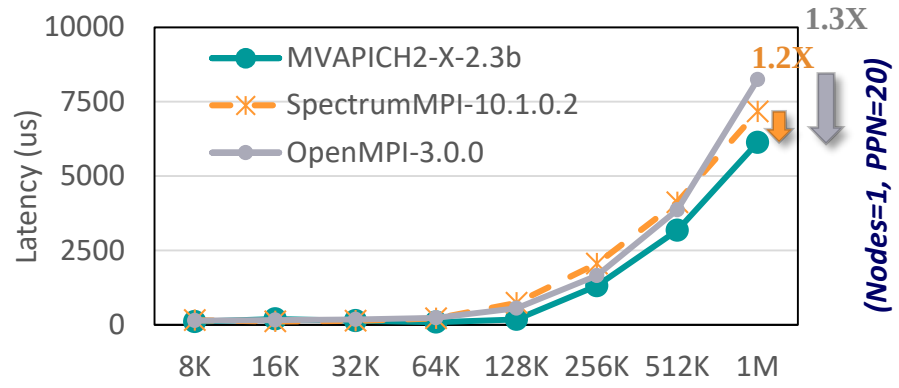
Reduce





Alltoall

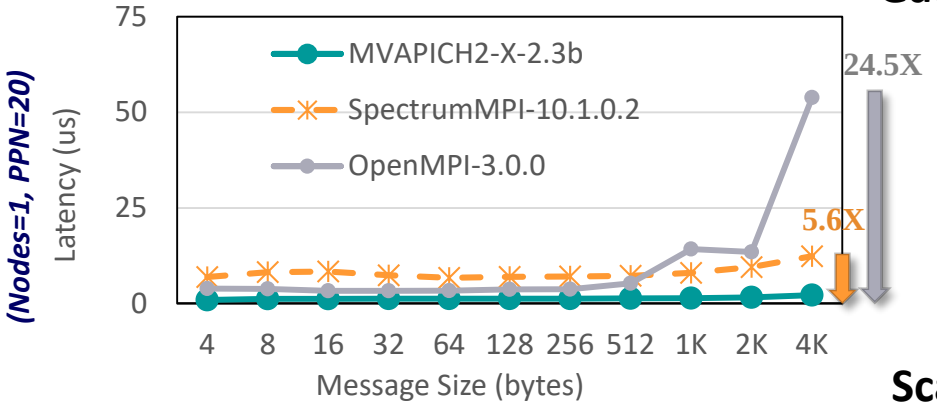


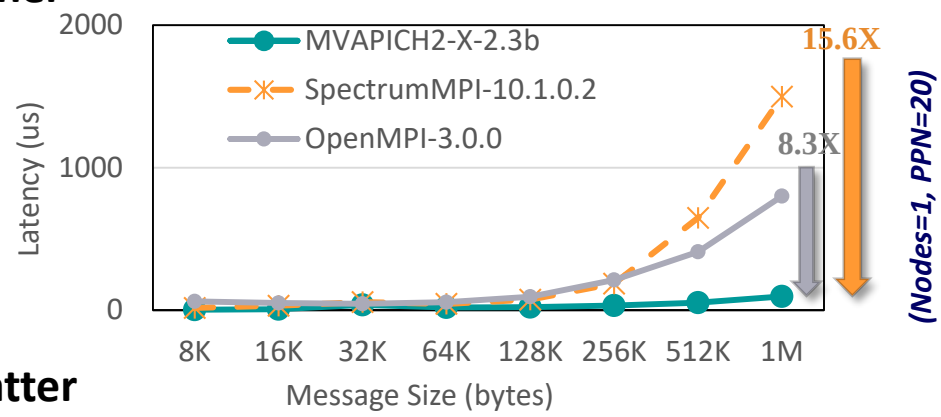


Up to 5X and 3x performance improvement by MVAPICH2-X-2.3b for small and large messages respectively

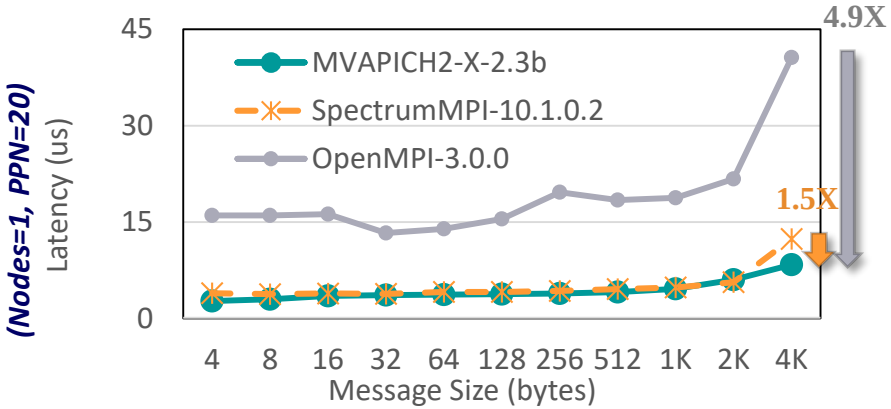
Scalable Collectives with CMA (Intra-node, Gather & Scatter)

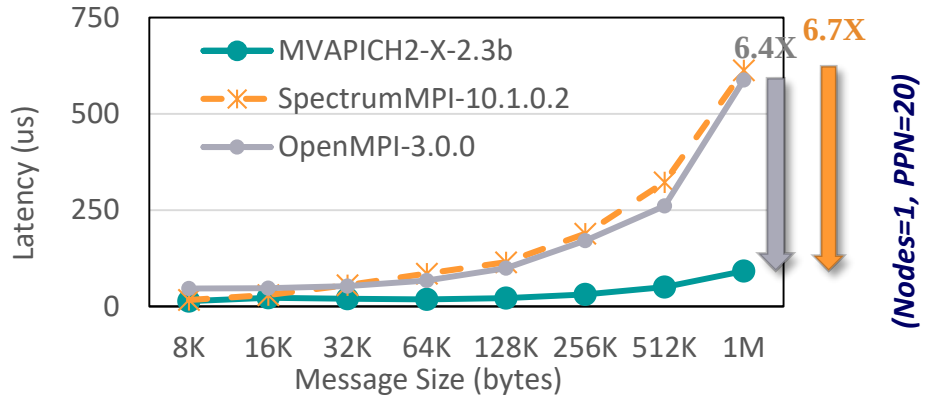
Gather





Scatter

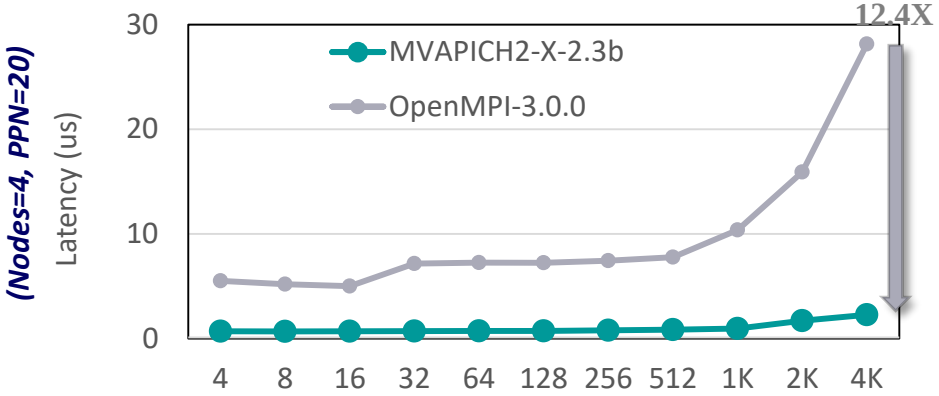


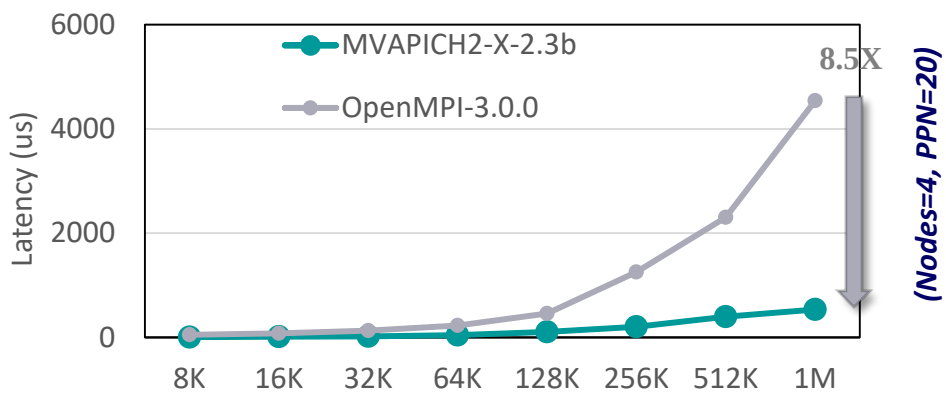


Up to 24X and 15x performance improvement by MVAPICH2-X-2.3b for medium to large messages respectively

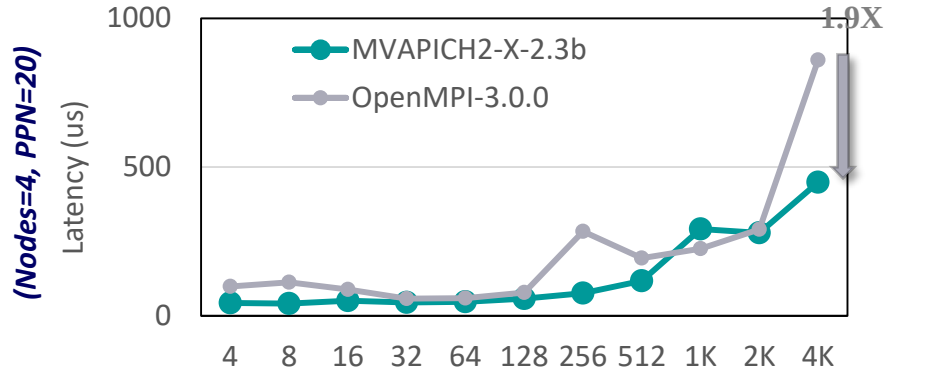
Scalable Collectives with CMA (Multi-node, Reduce & Alltoall)

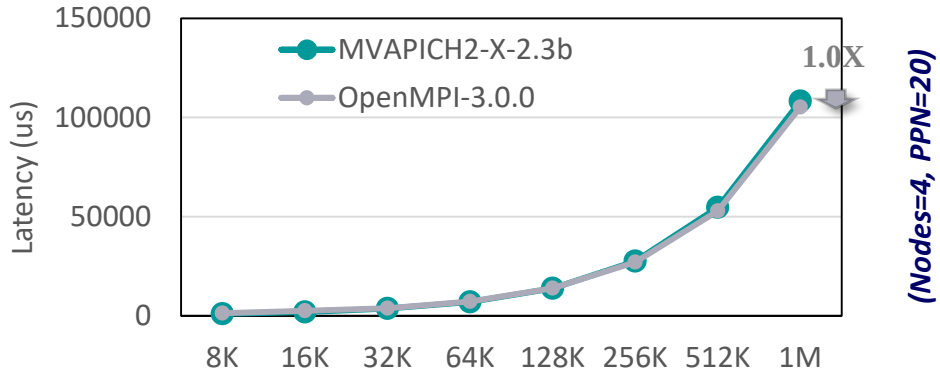
Reduce





Alltoall

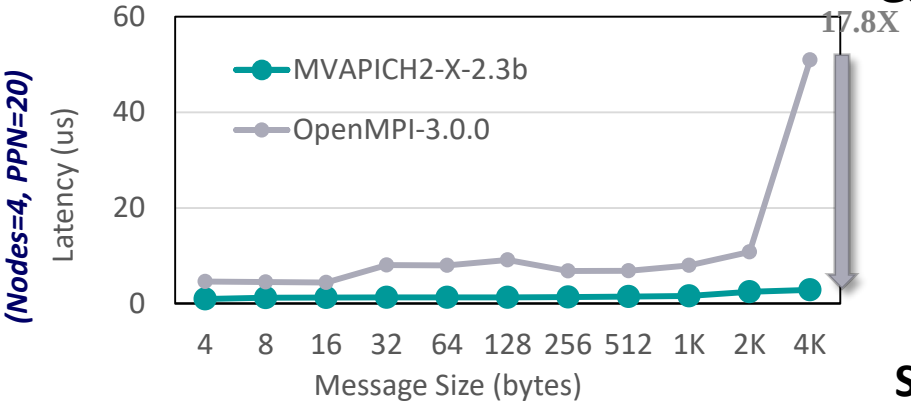


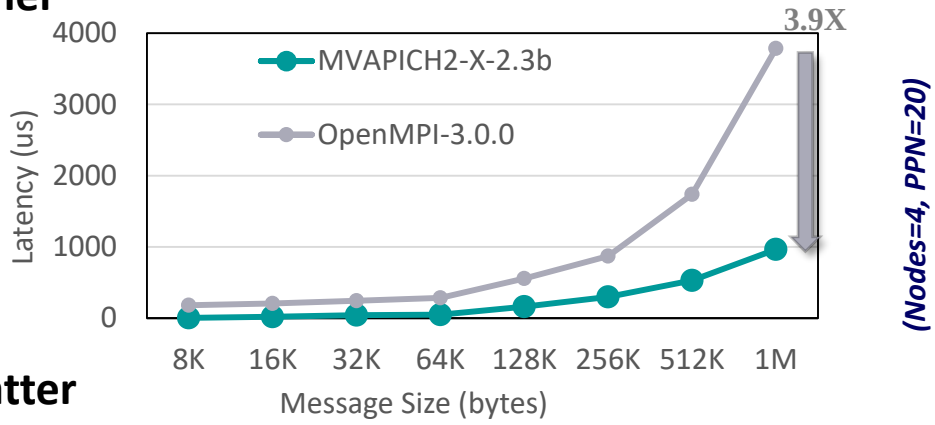


Up to 12.4X and 8.5X performance improvement by MVAPICH2-X-2.3b for small and large messages respectively

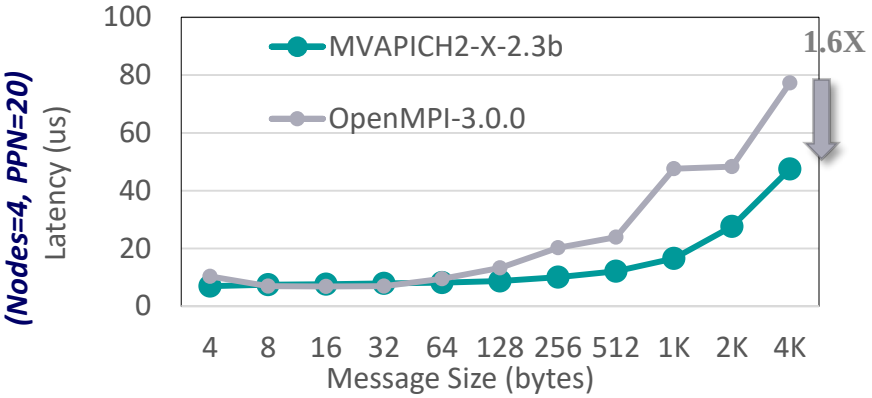
Scalable Collectives with CMA (Multi-node, Gather & Scatter)

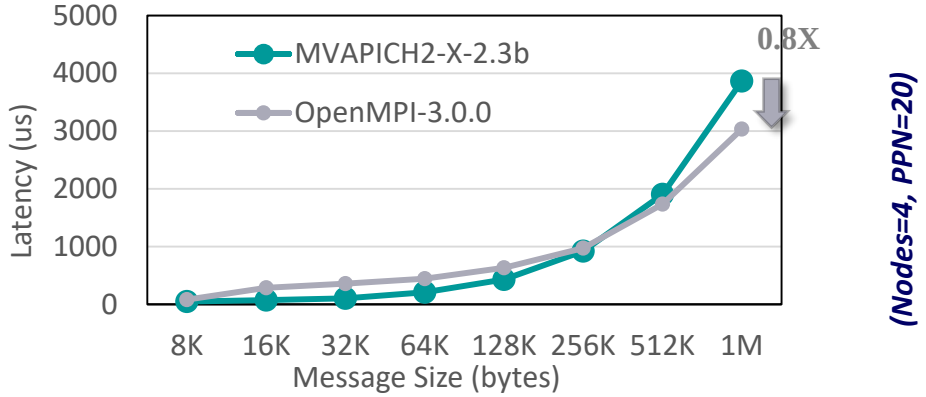
Gather





Scatter



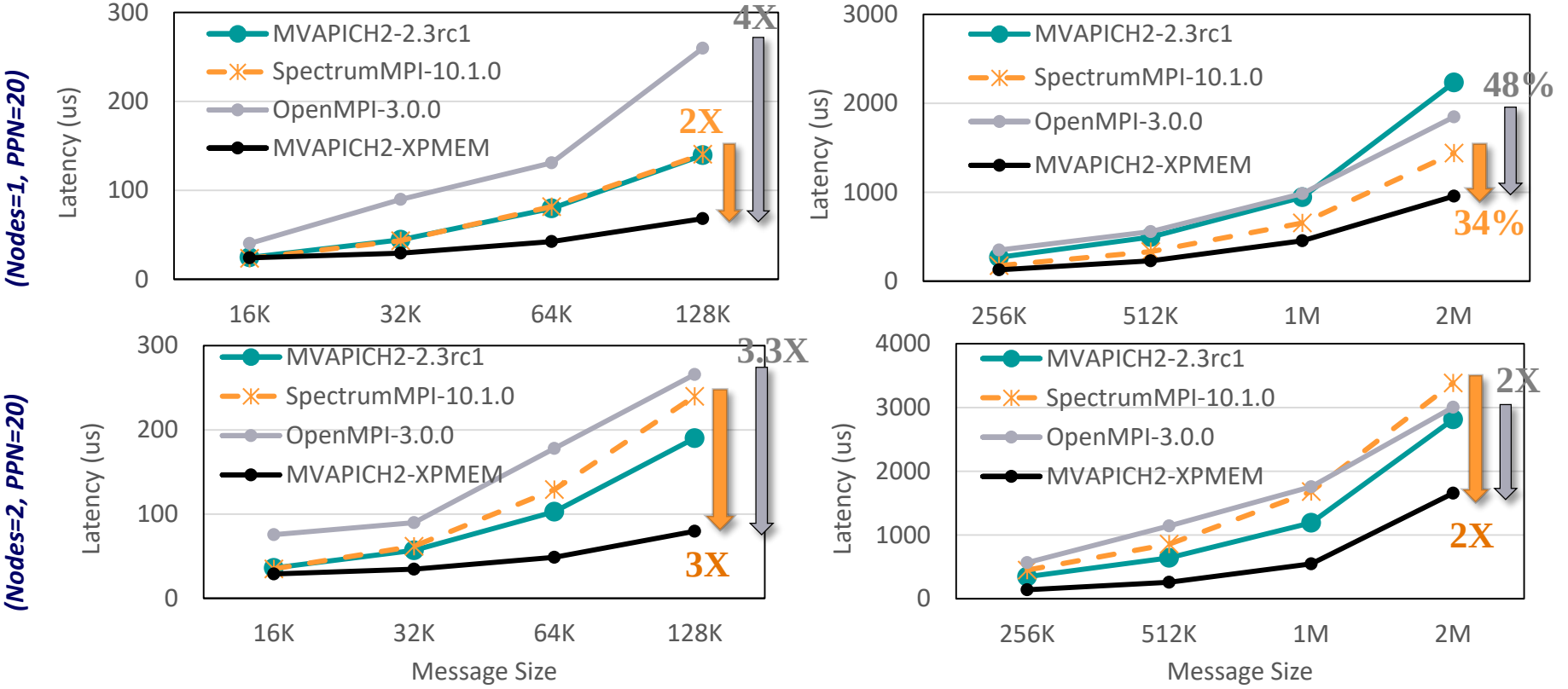


Up to 17.8X and 3.9X performance improvement by MVAPICH2-X-2.3b for medium to large messages respectively

MPI, PGAS and Deep Learning Support for OpenPOWER

- Message Passing Interface (MPI) Support
  - Point-to-point Inter-node and Intra-node
  - CMA-based Collectives
  - Upcoming XPMEM Support
- Support for PGAS and MPI + PGAS (OpenSHMEM, UPC)
- Exploiting Accelerators (NVIDIA GPGPUs)
- Support for Deep Learning

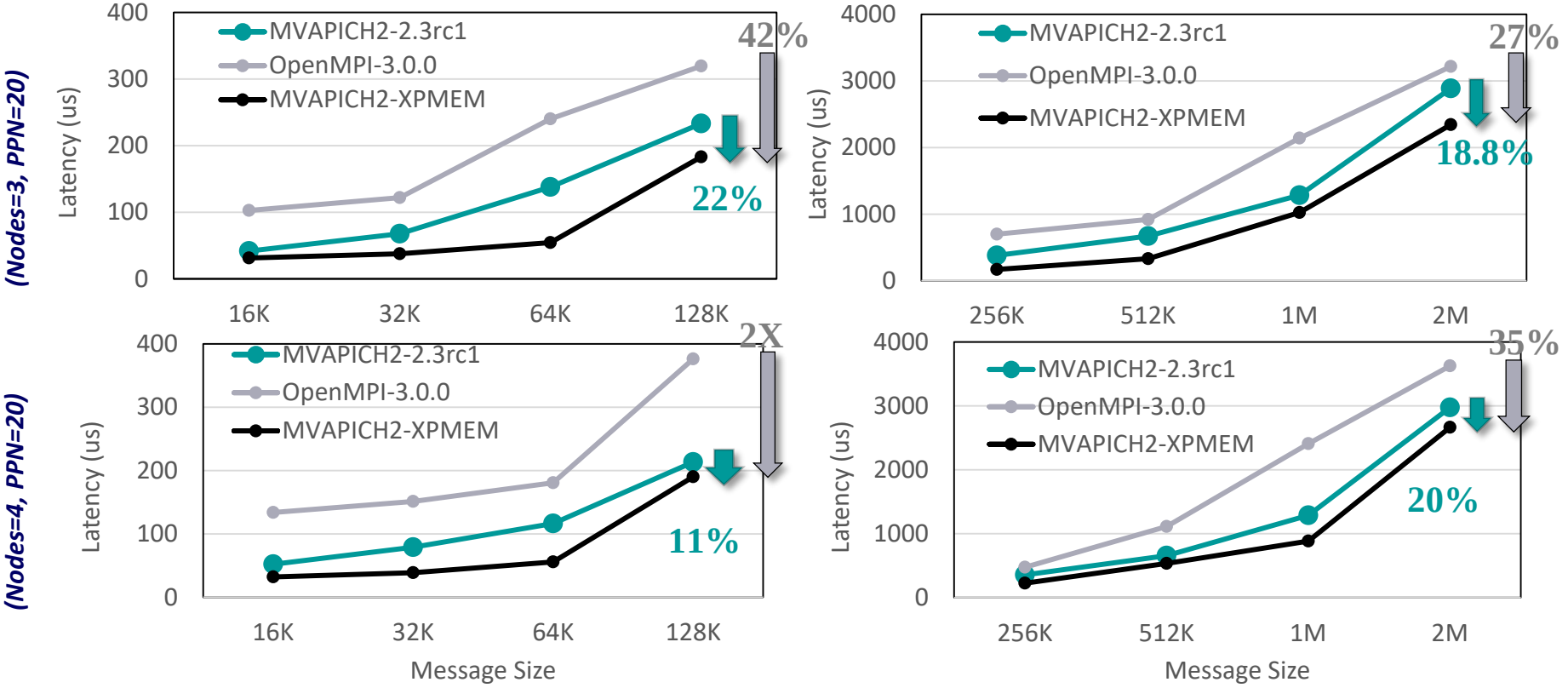
Optimized MVAPICH2 All-Reduce with XPMEM



- **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

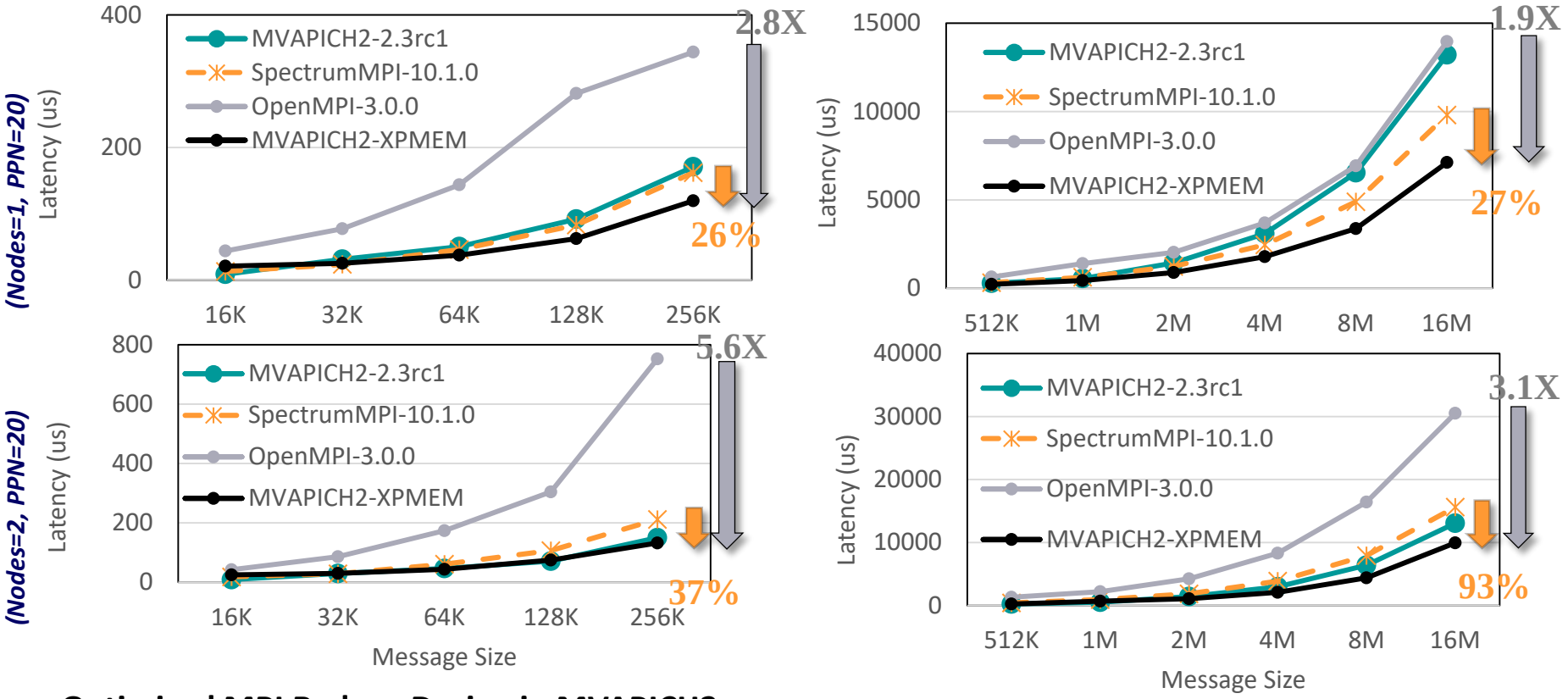
Optimized MVAPICH2 All-Reduce with XPMEM



- **Optimized MPI All-Reduce Design in MVAPICH2**
  - ***Up to 2X*** performance improvement over OpenMPI for inter-node. (Spectrum MPI didn't run for >2 processes)

Optimized Runtime Parameters: MV2_CPU_BINDING_POLICY=hybrid MV2_HYBRID_BINDING_POLICY=bunch

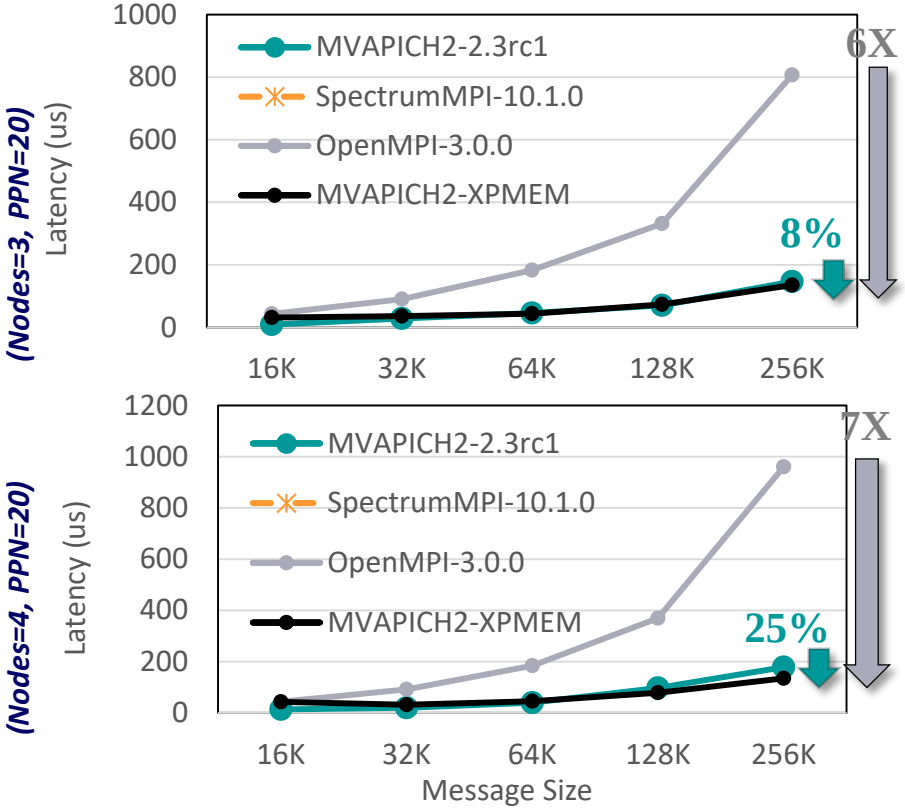
Optimized MVAPICH2 Reduce with XPMEM

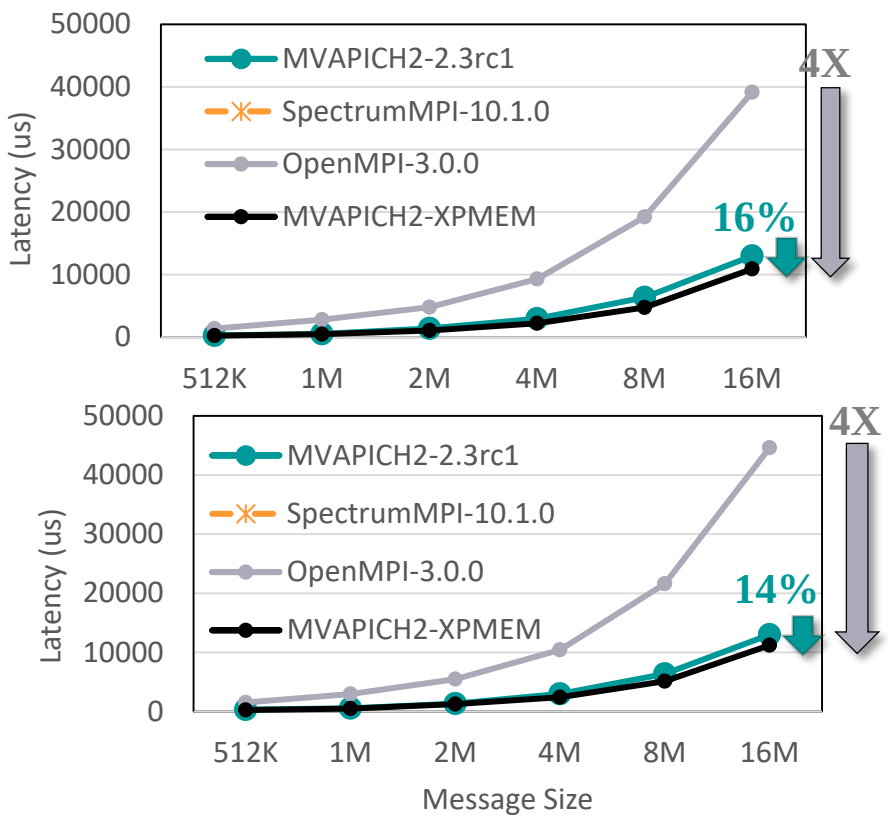


- **Optimized MPI Reduce Design in MVAPICH2**
  - ***Up to 3.1X*** performance improvement over OpenMPI at scale and up to **37%** over spectrum MPI on intra-node

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

Optimized MVAPICH2 Reduce with XPMEM

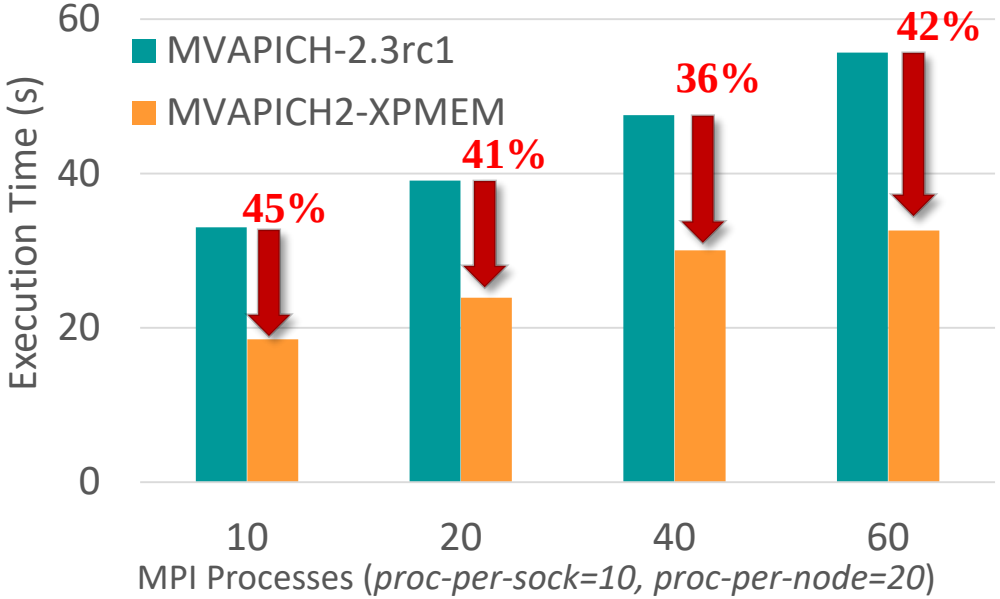




- **Optimized MPI Reduce Design in MVAPICH2**
  - ***Up to 7X*** performance improvement over OpenMPI at scale and up to **25%** over MVAPICH2-2.3rc1

Optimized Runtime Parameters: MV2_CPU_BINDING_POLICY=hybrid MV2_HYBRID_BINDING_POLICY=bunch

MiniAMR Performance using Optimized XPMEM-based Collectives



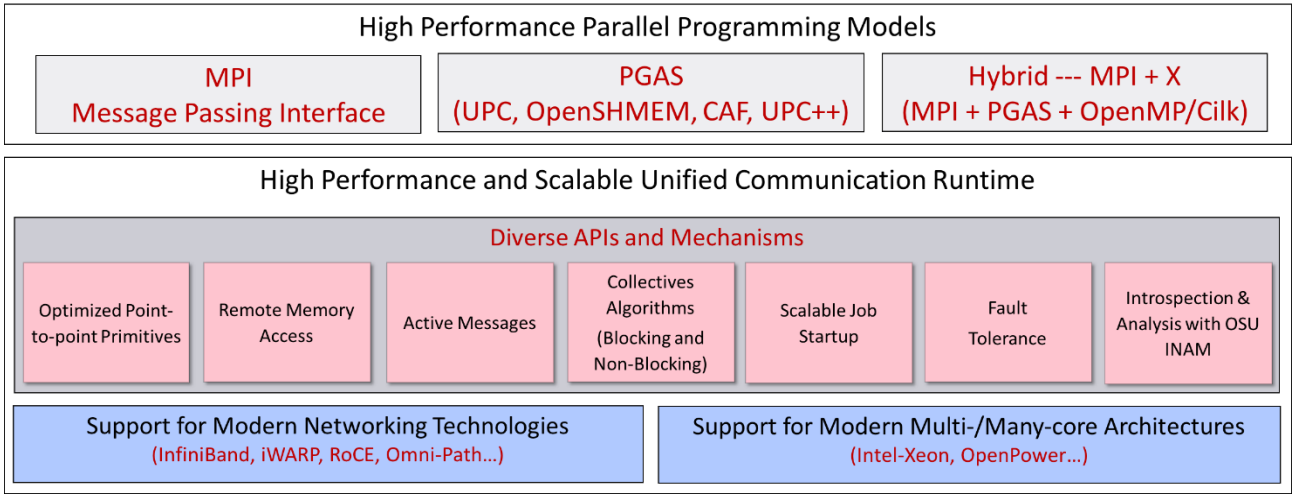
- **MiniAMR application execution time comparing MVAPICH2-2.3rc1 and optimized All-Reduce design**
  - ***Up to 45% improvement over MVAPICH2-2.3rc1 in mesh-refinement time of MiniAMR application for weak-scaling workload on up to four POWER8 nodes.***

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

MPI, PGAS and Deep Learning Support for OpenPOWER

- Message Passing Interface (MPI) Support
- Support for PGAS and MPI + PGAS (OpenSHMEM, UPC)
- Exploiting Accelerators (NVIDIA GPGPUs)
- Support for Deep Learning

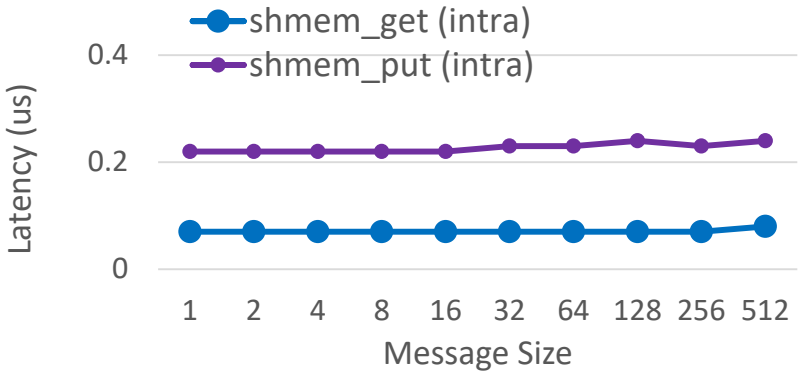
MVAPICH2-X for Hybrid MPI + PGAS Applications

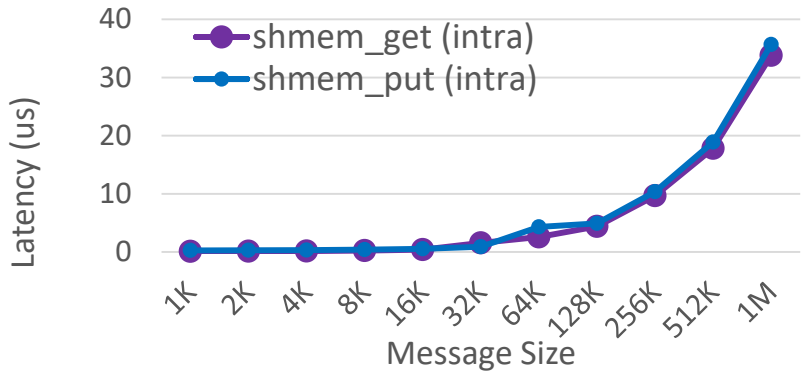


- Current Model – Separate Runtimes for OpenSHMEM/UPC/UPC++/CAF and MPI
  - Possible deadlock if both runtimes are not progressed
  - Consumes more network resource
- Unified communication runtime for MPI, UPC, UPC++, OpenSHMEM, CAF
  - Available with since 2012 (starting with MVAPICH2-X 1.9)
  - http://mvapich.cse.ohio-state.edu

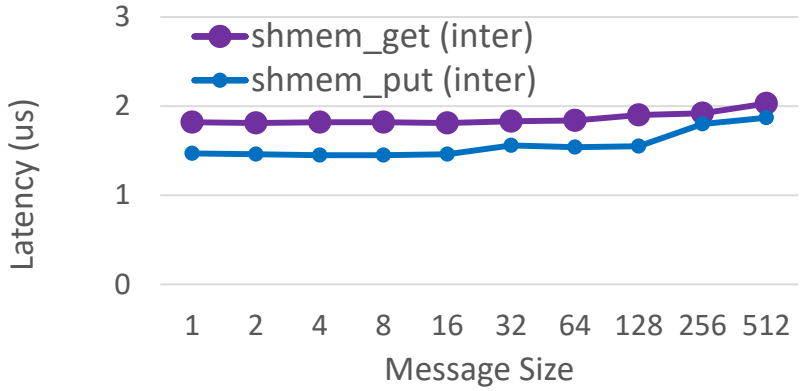
OpenSHMEM PUT/GET using MVAPICH2-X on OpenPOWER

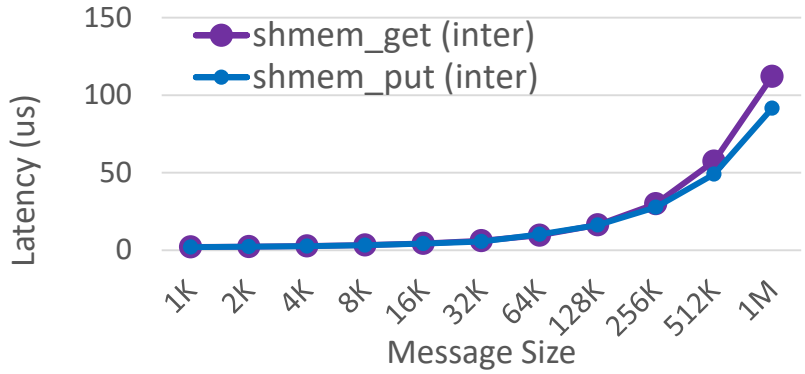
Intra-node (PPN=2)





Inter-node (PPN=1)

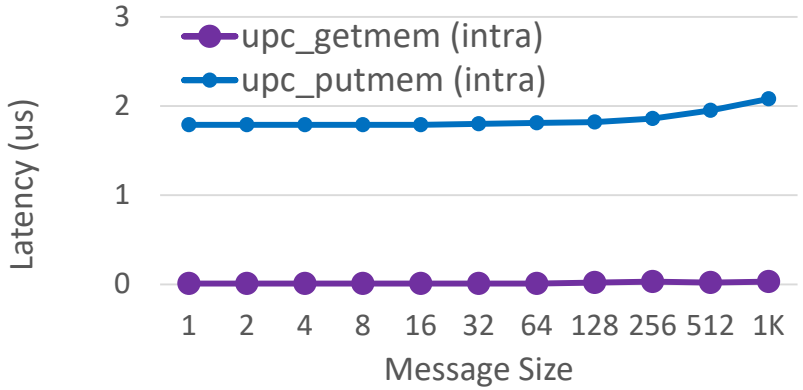


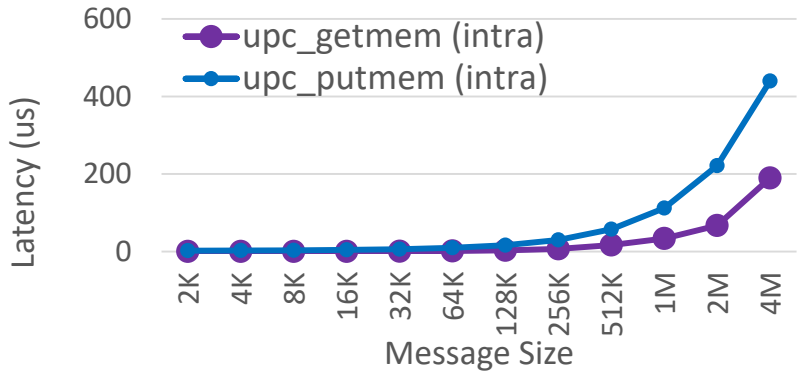


- **OpenSHMEM one-sided PUT/GET benchmark using MVAPICH2-X 2.3b on OpenPOWER cluster**
  - *Near native performance benefits are observed on OpenPOWER*

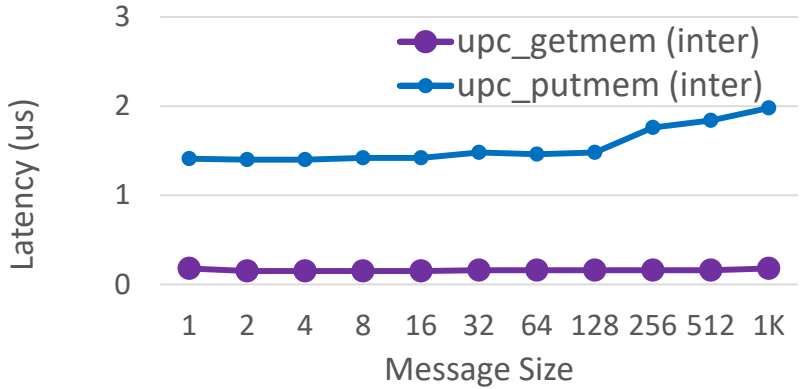
UPC PUT/GET benchmarks using MVAPICH2-X on OpenPOWER

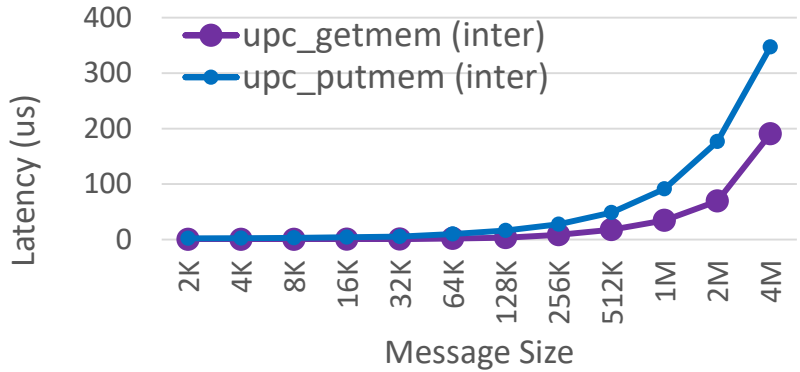
Intra-node (PPN=2)





Inter-node (PPN=1)

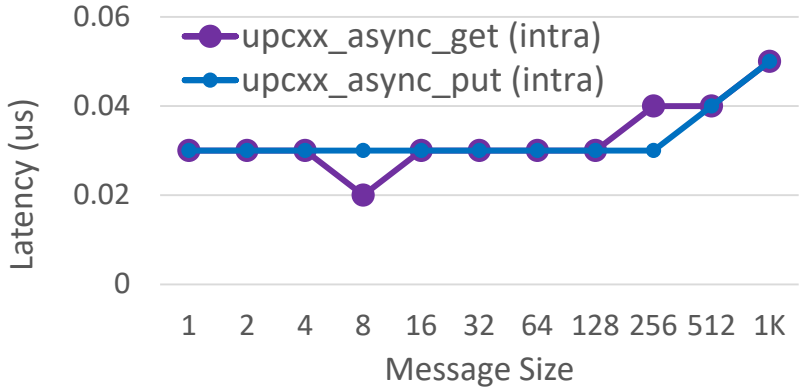


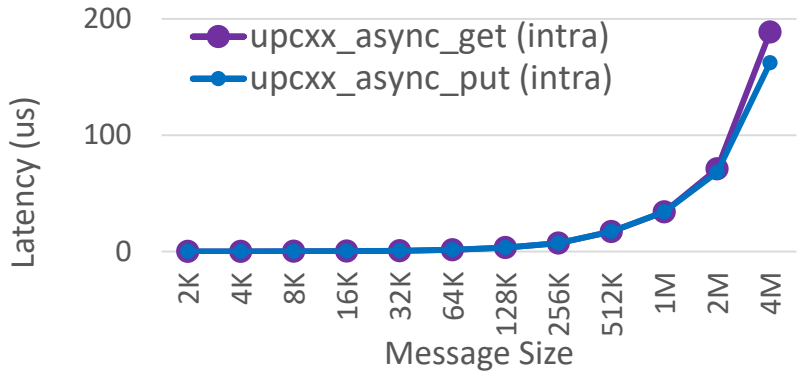


- **UPC one-sided memput/memget benchmarks using MVAPICH2-X 2.3b on OpenPOWER cluster**
  - *Near native performance benefits are observed on OpenPOWER*

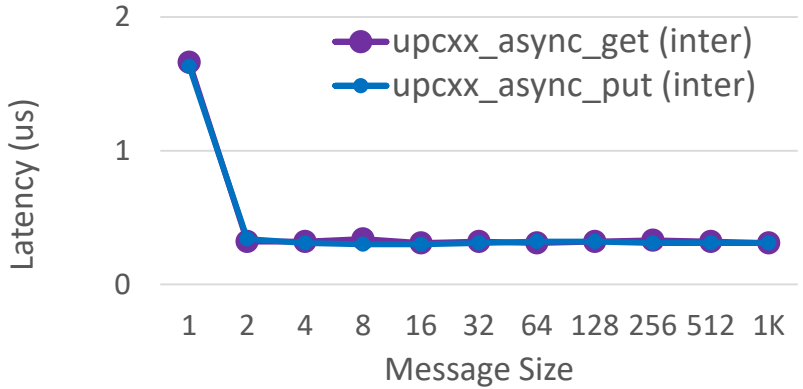
UPC++ PUT/GET benchmarks using MVAPICH2-X on OpenPOWER

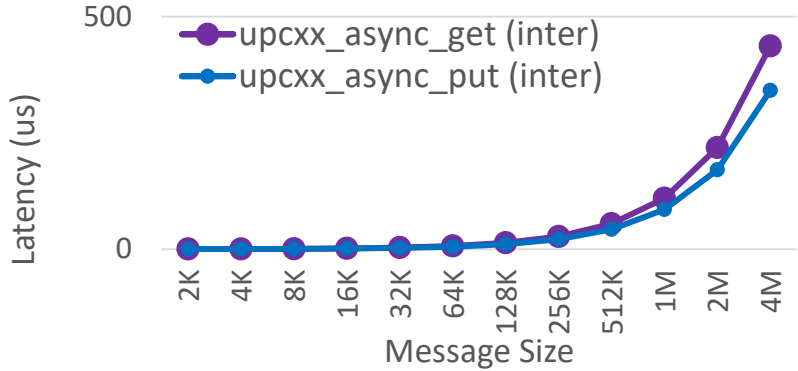
Intra-node (PPN=2)





Inter-node (PPN=1)





- **UPC++ one-sided async_copy_put/get benchmarks using MVAPICH2-X 2.3b on OpenPOWER cluster**
  - *Near native performance benefits are observed on OpenPOWER*

MPI, PGAS and Deep Learning Support for OpenPOWER

- Message Passing Interface (MPI) Support
- Support for PGAS and MPI + PGAS (OpenSHMEM, UPC)
- Exploiting Accelerators (NVIDIA GPGPUs)
  - CUDA-aware MPI
  - Optimized GPUDirect RDMA (GDR) Support
  - Optimized MVAPICH2 for OpenPower
- Support for Deep Learning

GPU-Aware (CUDA-Aware) MPI Library: MVAPICH2-GPU

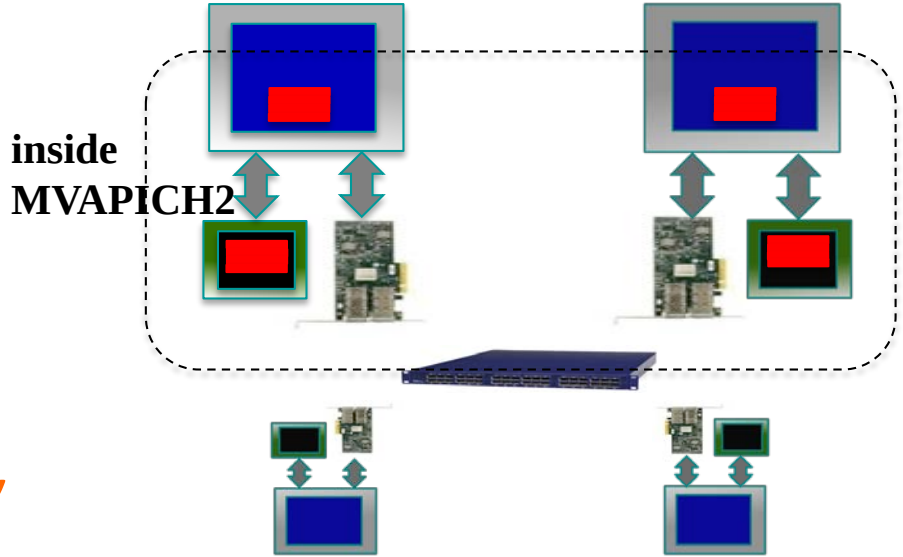
- Standard MPI interfaces used for unified data movement
- Takes advantage of Unified Virtual Addressing (>= CUDA 4.0)
- Overlaps data movement from GPU with RDMA transfers

At Sender:

```
MPI_Send(s_devbuf, size, …);
```

At Receiver:

```
MPI_Recv(r_devbuf, size, …);
```

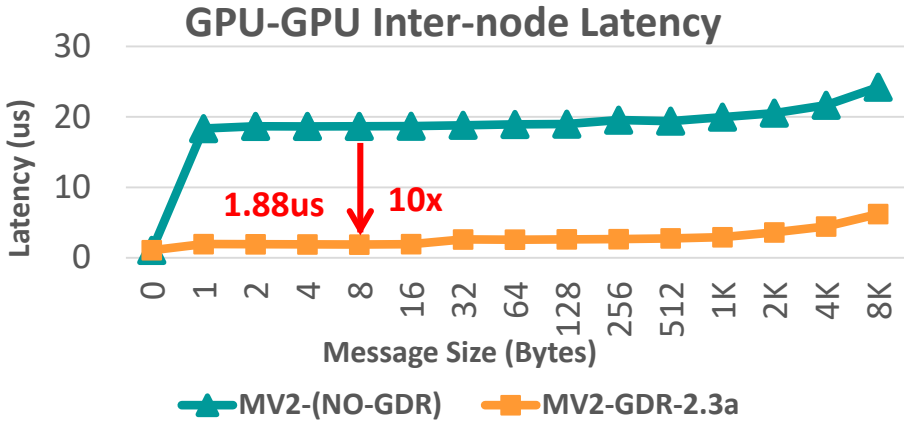


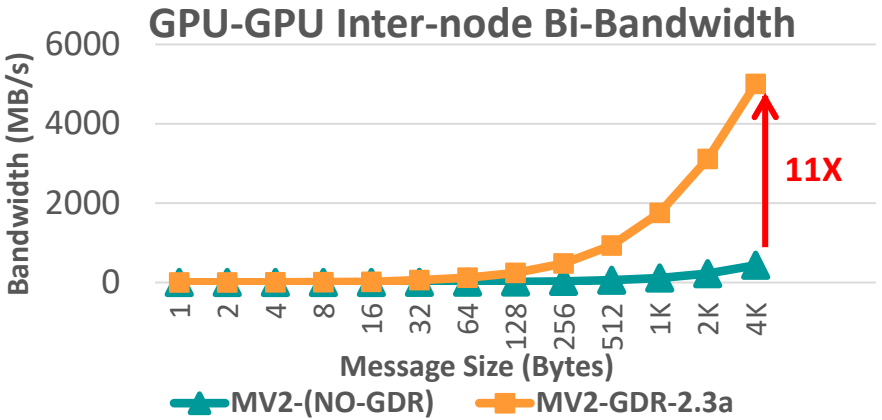
High Performance and High Productivity

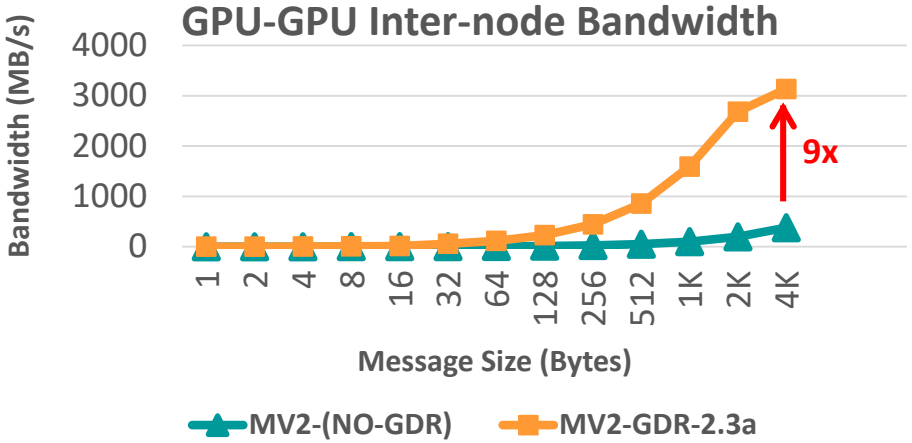
CUDA-Aware MPI: MVAPICH2-GDR 1.8-2.3 Releases

- 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

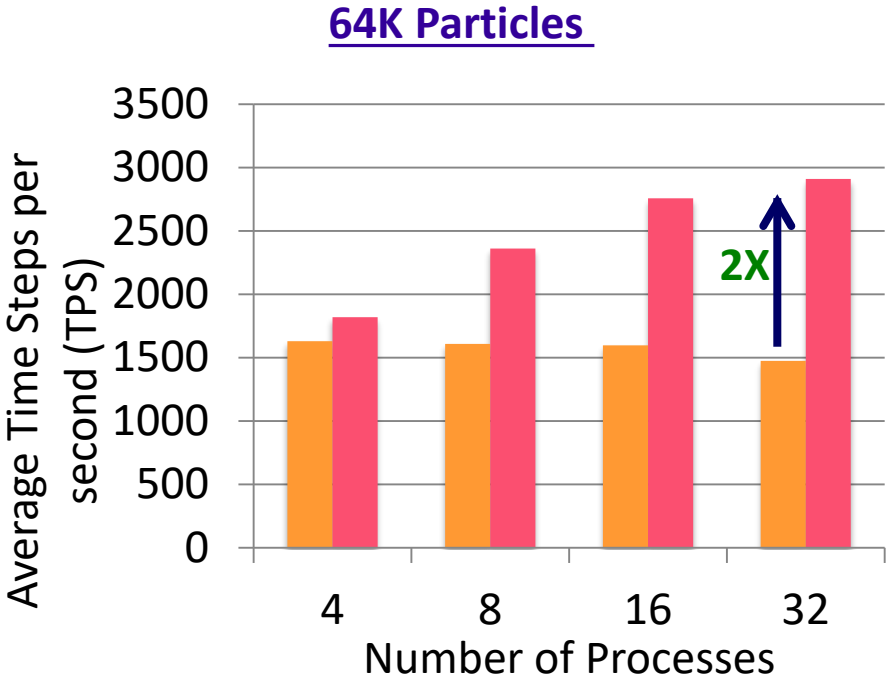


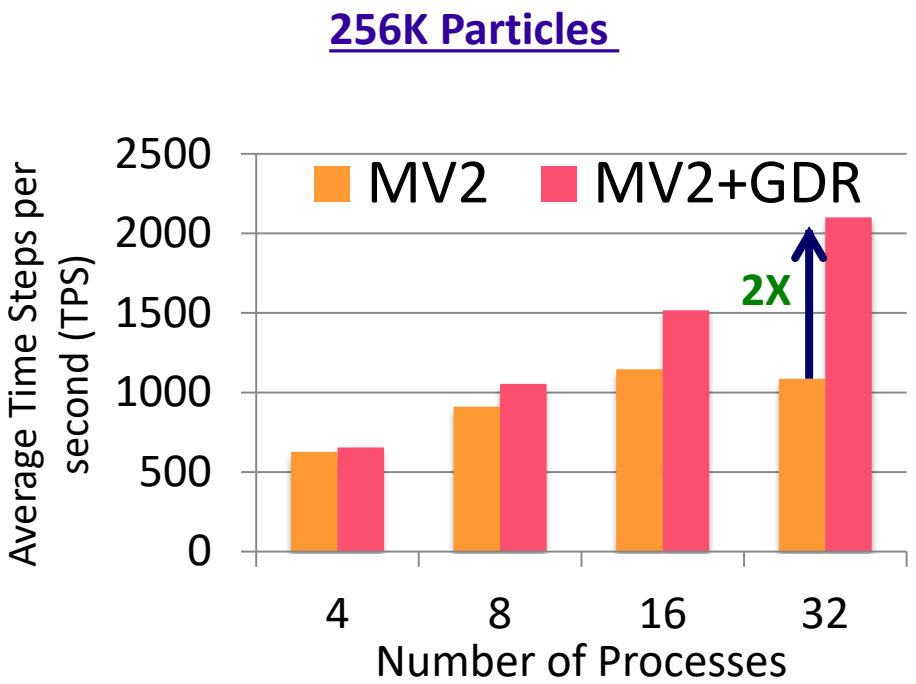




MVAPICH2-GDR-2.3a
Intel Haswell (E5-2687W @ 3.10 GHz) node - 20 cores
NVIDIA Volta V100 GPU
Mellanox Connect-X4 EDR HCA
CUDA 9.0
Mellanox OFED 4.0 with GPU-Direct-RDMA

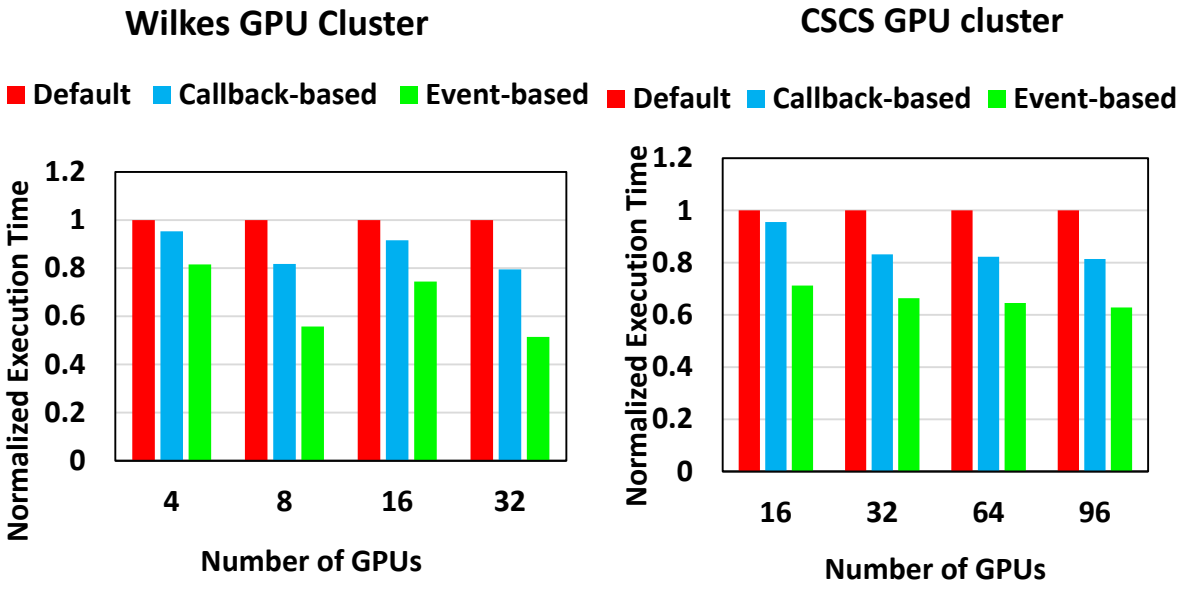
Application-Level Evaluation (HOOMD-blue)

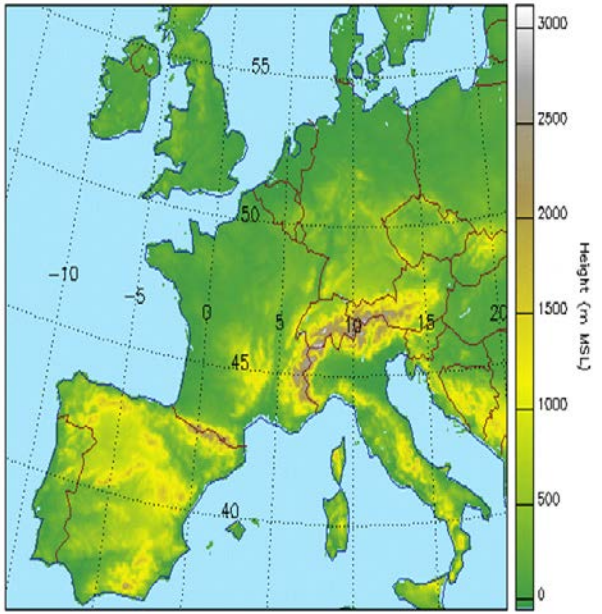




- Platform: Wilkes (Intel Ivy Bridge + NVIDIA Tesla K20c + Mellanox Connect-IB)
- HoomDBlue Version 1.0.5
  - GDRCOPY enabled: MV2_USE_CUDA=1 MV2_IBA_HCA=mlx5_0 MV2_IBA_EAGER_THRESHOLD=32768 MV2_VBUF_TOTAL_SIZE=32768 MV2_USE_GPUDIRECT_LOOPBACK_LIMIT=32768 MV2_USE_GPUDIRECT_GDRCOPY=1 MV2_USE_GPUDIRECT_GDRCOPY_LIMIT=16384

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





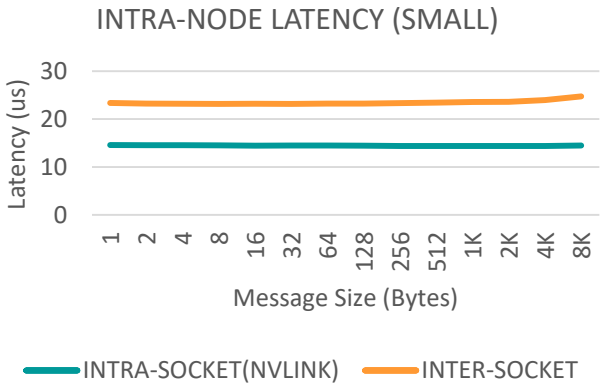
- 2X improvement on 32 GPUs nodes
- 30% improvement on 96 GPU nodes (8 GPUs/node)

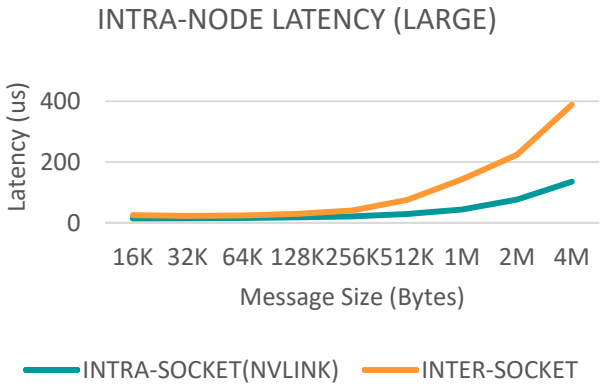
Cosmo model: http://www2.cosmo-model.org/content/tasks/operational/meteoSwiss/

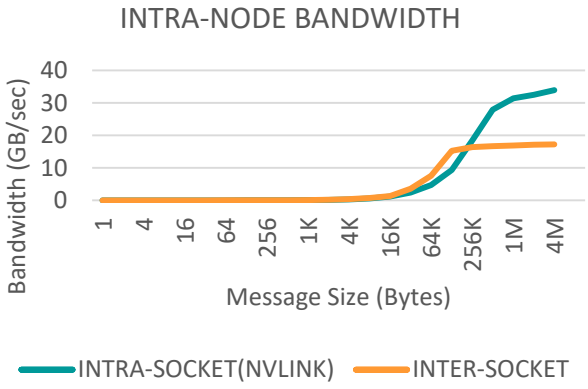
On-going collaboration with CSCS and MeteoSwiss (Switzerland) in co-designing MV2-GDR and Cosmo Application

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-GDR: Performance on OpenPOWER (NVLink + Pascal)

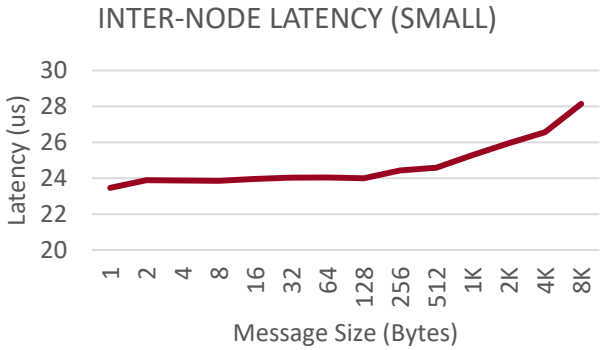


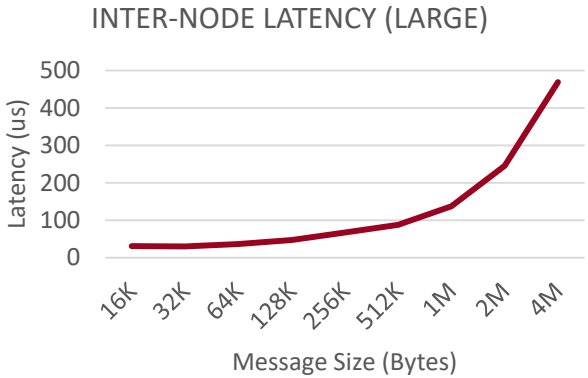


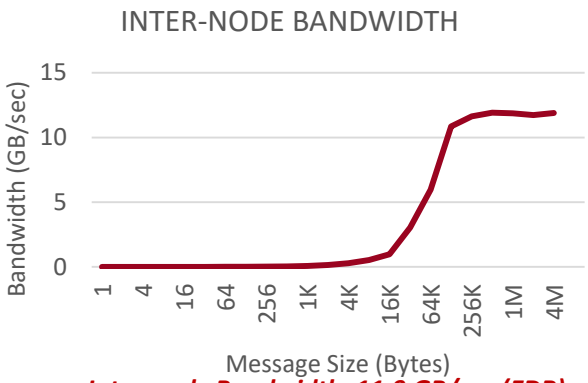


Intra-node Latency: 14.6 us (without GPUDirectRDMA)

Intra-node Bandwidth: 33.9 GB/sec (NVLINK)







Inter-node Latency: 23.8 us (without GPUDirectRDMA)

Available in MVAPICH2-GDR 2.3a

Inter-node Bandwidth: 11.9 GB/sec (EDR)

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

Compilation of Best Practices

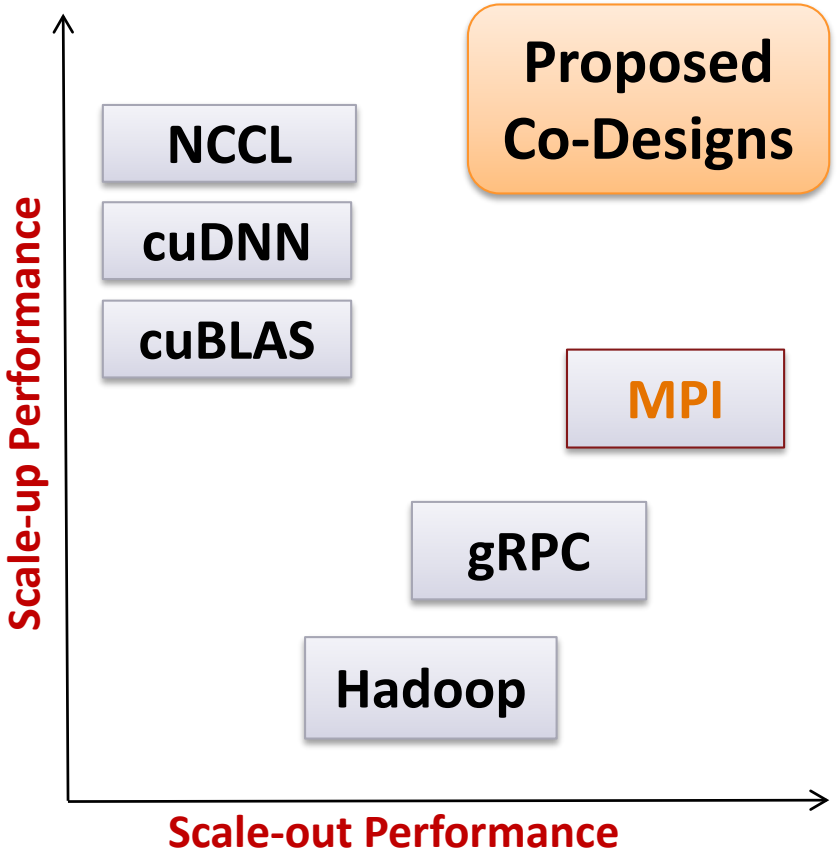
- MPI runtime has many parameters
- Tuning a set of parameters can help you to extract higher performance
- Compiled a list of such contributions through the MVAPICH Website
  - http://mvapich.cse.ohio-state.edu/best_practices/
- List of applications (16 contributions so far)
  - Amber, Cloverleaf, GAPgeofem, HoomDBlue, HPCG, LAMPS, LU, Lulesh, MILC, Neuro, POP2, SMG2000, SPEC MPI, TERA_TF, UMR, and WRF2
- Soliciting additional contributions, send your results to mvapich-help at cse.ohio-state.edu
- We will link these results with credits to you

MPI, PGAS and Deep Learning Support for OpenPOWER

- Message Passing Interface (MPI) Support
- Support for PGAS and MPI + PGAS (OpenSHMEM, UPC)
- Exploiting Accelerators (NVIDIA GPGPUs)
- Support for Deep Learning
  - New challenges for MPI Run-time
  - Optimizing Large-message Collectives
  - Co-designing DL Frameworks

Deep Learning: New Challenges for MPI Runtimes

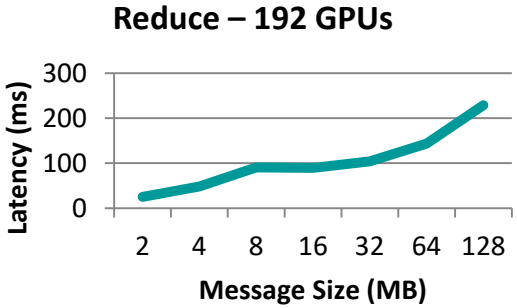
- Deep Learning frameworks are a different game altogether
  - Unusually large message sizes (order of megabytes)
  - Most communication based on GPU buffers
- Existing State-of-the-art
  - cuDNN, cuBLAS, NCCL --> **scale-up** performance
  - CUDA-Aware MPI --> **scale-out** performance
    - For small 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**?

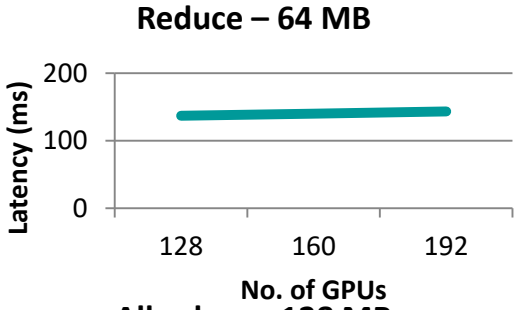


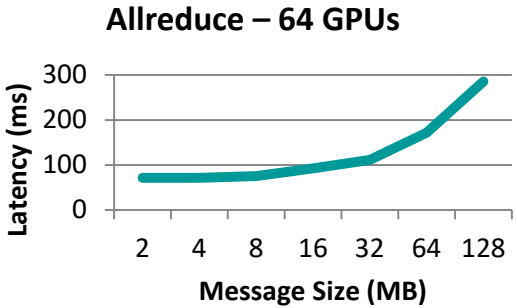
A. A. Awan, K. Hamidouche, J. M. Hashmi, and D. K. Panda, S-Caffe: Co-designing MPI Runtimes and Caffe for Scalable Deep Learning on Modern GPU Clusters. In *Proceedings of the 22nd ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming* (PPoPP '17)

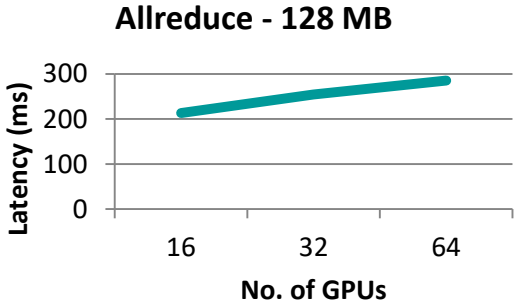
Large Message Optimized Collectives for Deep Learning

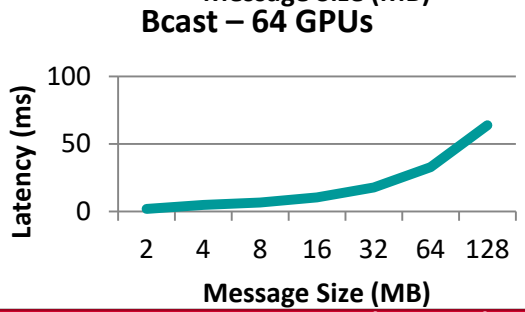
- MV2-GDR provides optimized collectives for **large message sizes**
- Optimized Reduce, Allreduce, and Bcast
- **Good scaling with large number of GPUs**
- **Available since MVAPICH2-GDR 2.2GA**

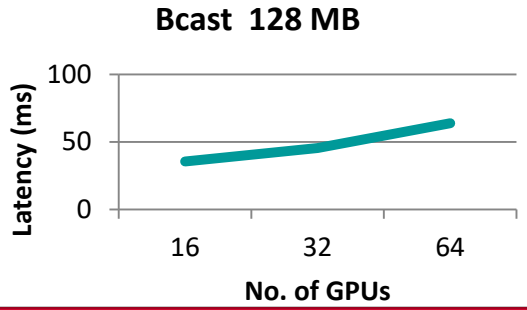






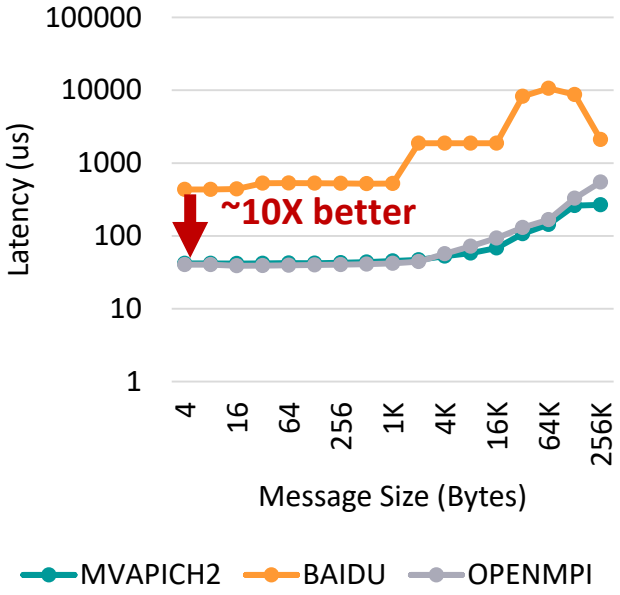


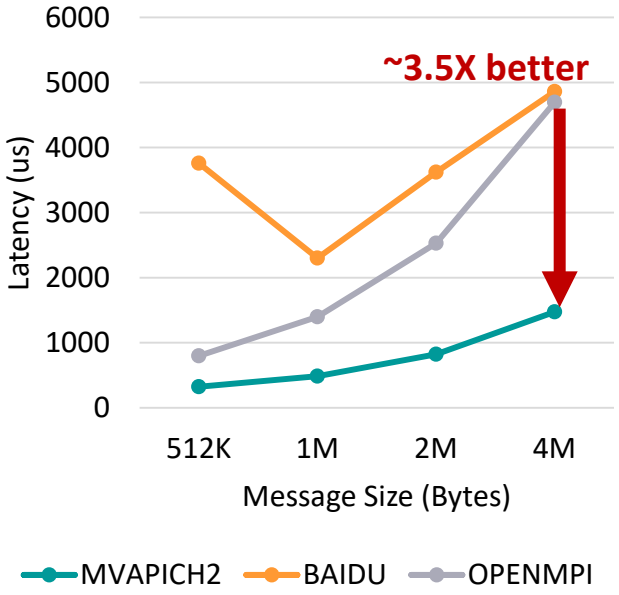


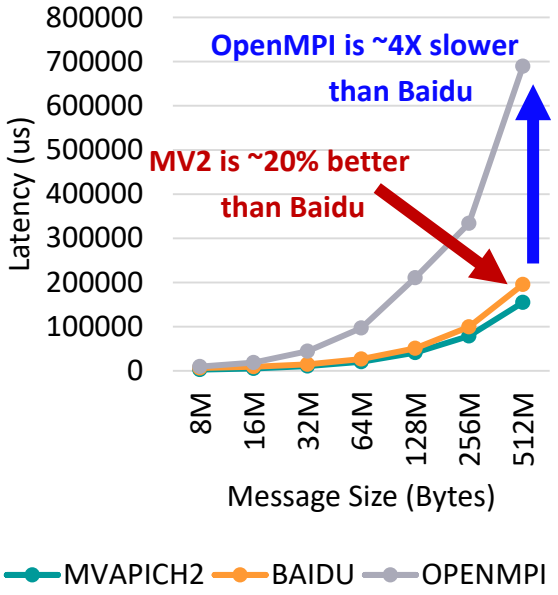


MVAPICH2: Allreduce Comparison with Baidu and OpenMPI

- Initial Evaluation shows promising performance gains for MVAPICH2-GDR 2.3a*
- 8 GPUs (2 nodes) MVAPICH2-GDR vs. Baidu-Allreduce and OpenMPI 3.0





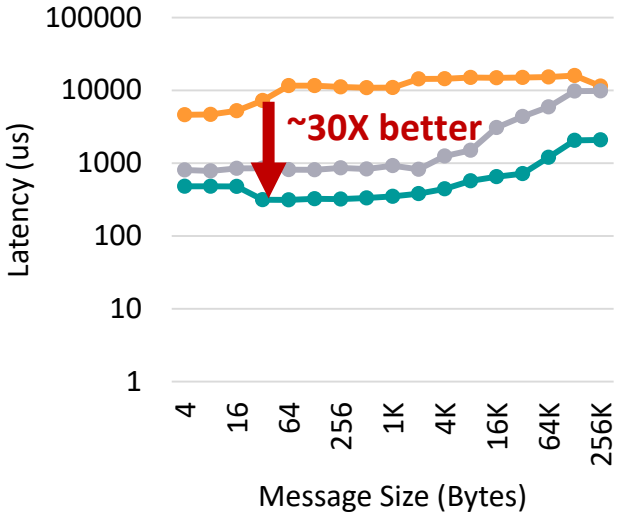


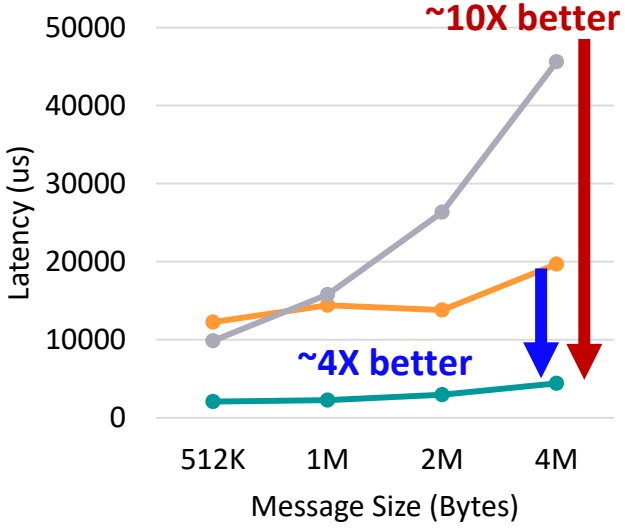
**Available with MVAPICH2-GDR 2.3a and higher*

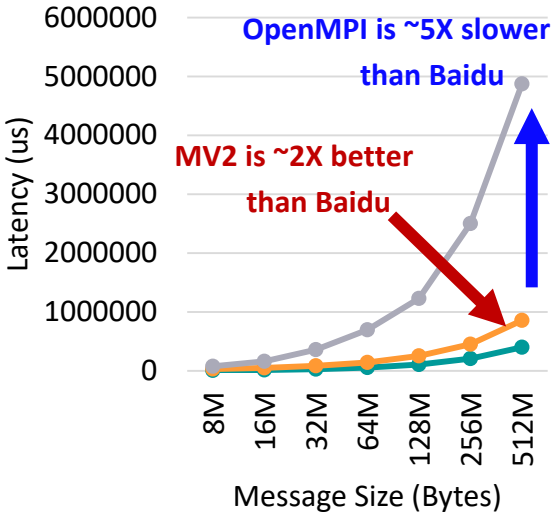
Platform: OpenPOWER (ppc64le) nodes equipped with a dual-socket CPU, 4 Pascal P100-SXM GPUs, and FDR InfiniBand Inter-connect

MVAPICH2: Allreduce Comparison with Baidu and OpenMPI

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





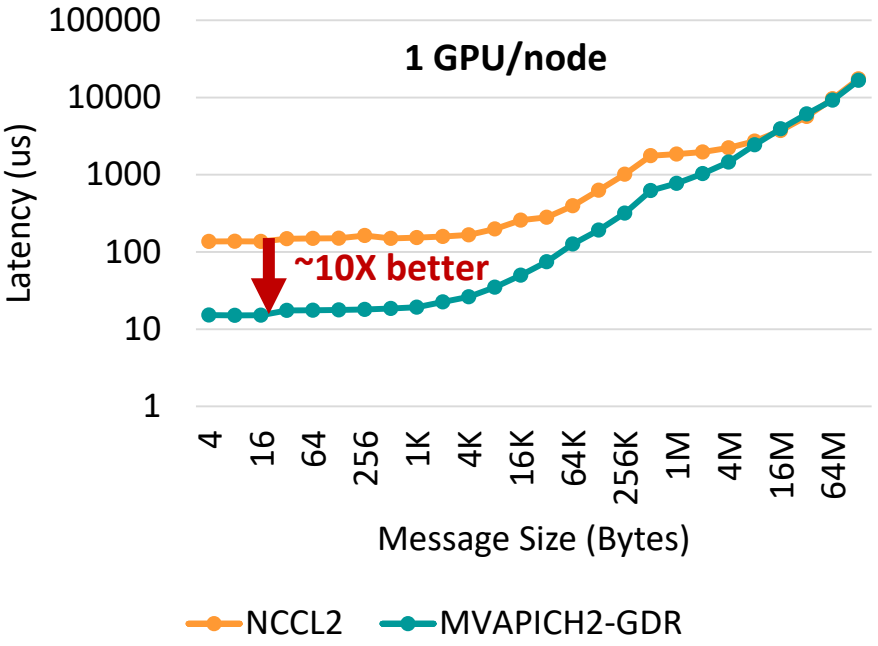


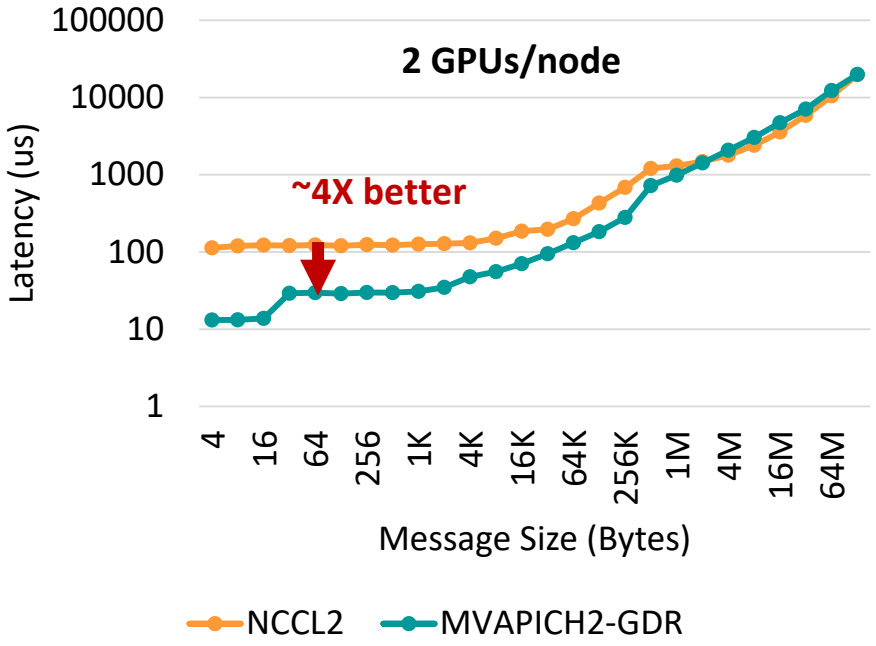
**Available with MVAPICH2-GDR 2.3a and higher*

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

MVAPICH2-GDR vs. NCCL2

- Optimized designs in MVAPICH2-GDR 2.3b* offer better/comparable performance for most cases
- MPI_Bcast (MVAPICH2-GDR) vs. ncclBcast (NCCL2) on 16 K-80 GPUs





*Will be available with upcoming MVAPICH2-GDR 2.3b

Platform: Intel Xeon (Broadwell) nodes equipped with a dual-socket CPU, 2 K-80 GPUs, and EDR InfiniBand Inter-connect

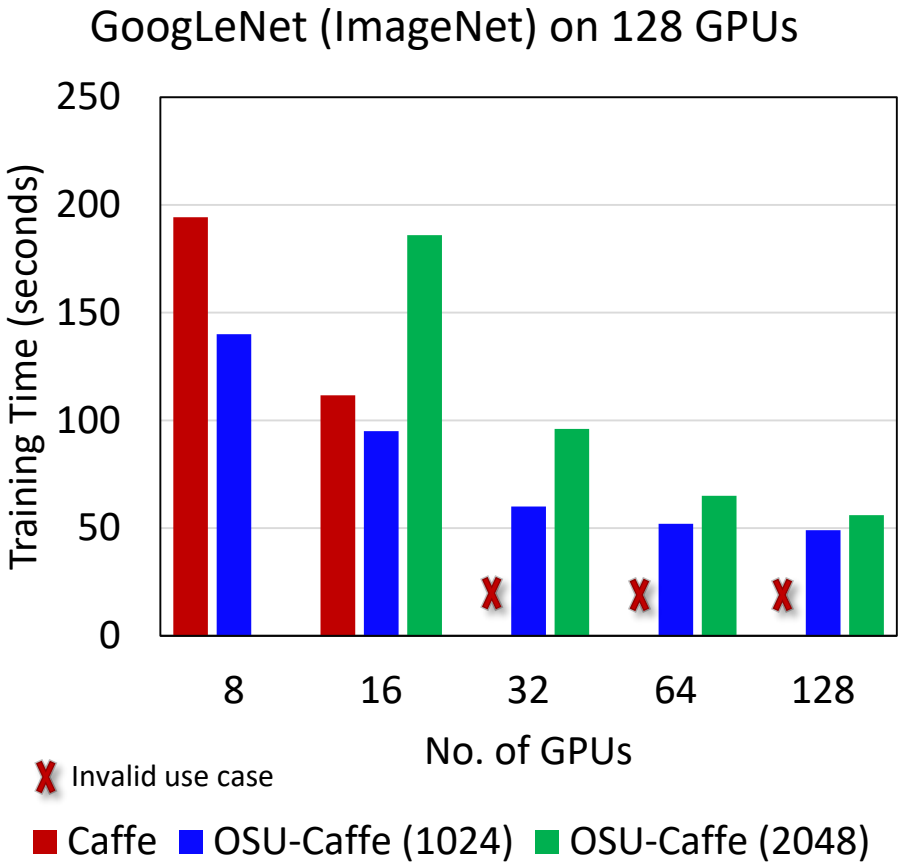
OSU-Caffe: Scalable Deep Learning

- 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/

Support on OPENPOWER will be available soon



Concluding Remarks

- Next generation HPC systems need to be designed with a holistic view of Big Data and Deep Learning
- OpenPOWER platform is emerging with many novel features
- Presented some of the approaches and results along these directions from the MVAPICH2 and HiDL Projects
- Solutions Enable High-Performance and Scalable HPC and Deep Learning on OpenPOWER Platforms

High-Performance Hadoop and Spark on OpenPOWER Platform

Talk at 2:15 pm

Funding Acknowledgments

Funding Support by

Equipment Support by

Personnel Acknowledgments

Current Students (Graduate)

- A. Awan (Ph.D.)
- R. Biswas (M.S.)
- 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.)
- J. Zhang (Ph.D.)

Current Students (Undergraduate)

- N. Sarkauskas (B.S.)

Current Research Scientists

- X. Lu
- H. Subramoni

Current Research Specialist

- J. Smith
- M. Arnold

Current Post-doc

- A. Ruhela

Past Students

- A. Augustine (M.S.)
- P. Balaji (Ph.D.)
- 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.)

Past Research Scientist

- K. Hamidouche
- S. Sur

Past Post-Docs

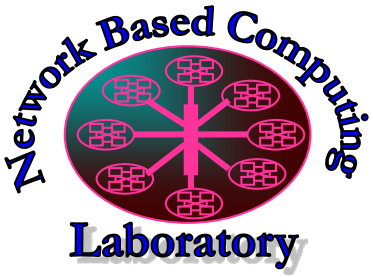
- D. Banerjee
- X. Besseron
- H.-W. Jin
- J. Lin
- M. Luo
- E. Mancini
- S. Marcarelli
- J. Vienne
- H. Wang

Past Programmers

- D. Bureddy
- J. Perkins

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/