# The MVAPICH2 Project: Latest Developments and Plans Towards Exascale Computing

Presentation at OSU Booth (SC '22)

by

**Hari Subramoni**

The Ohio State University

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

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

# Outline

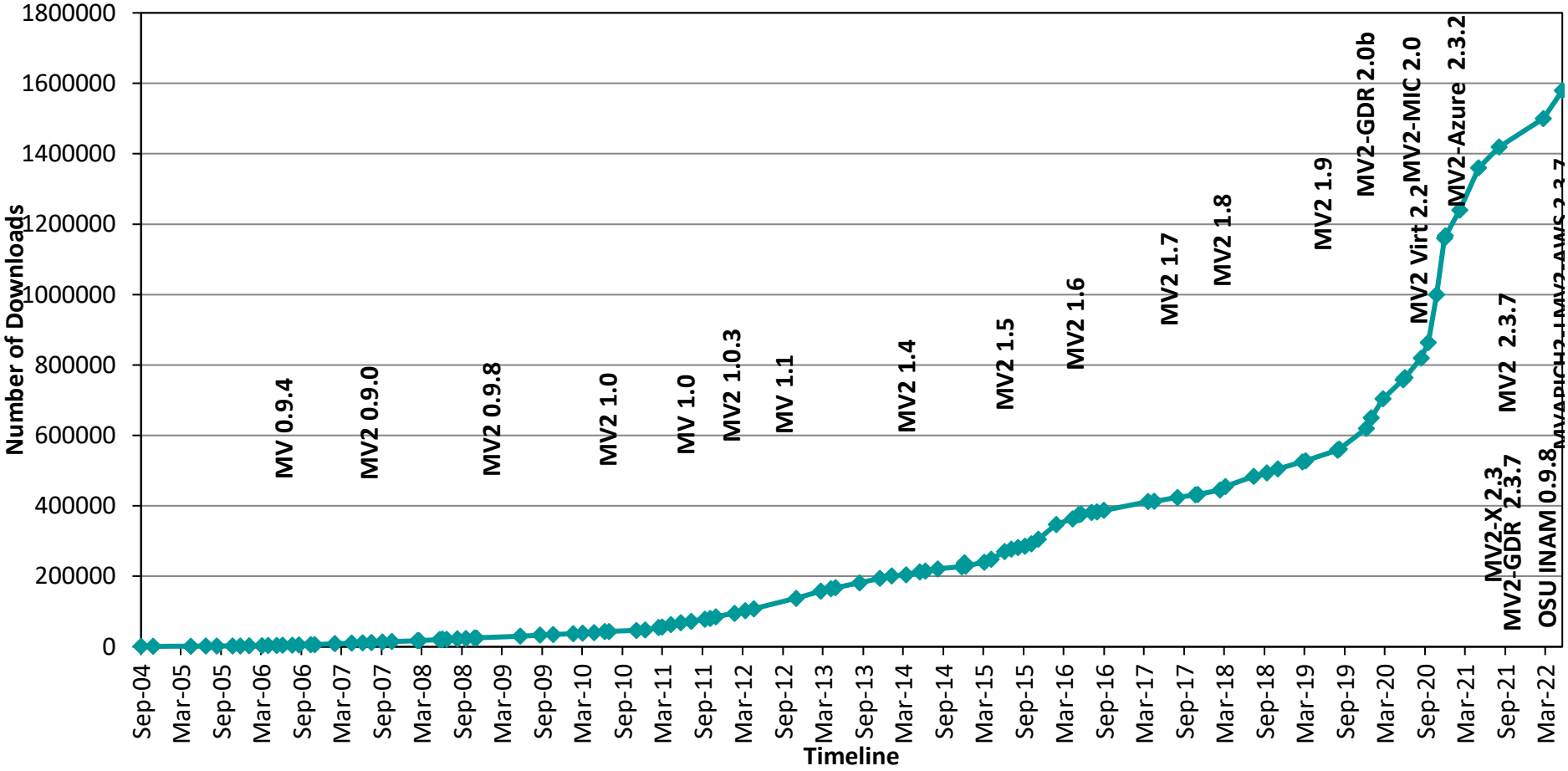
- **Brief Overview of the MVAPICH2 Project**
- Features of Recent Releases
  - MVAPICH2 2.3.7
  - MVAPICH2 3.0a
  - MVAPICH2-J
  - MVAPICH2-X-AWS 2.3.7 and Cloud Deployments
  - MVAPICH2-GDR 2.3.7 and Support for ML, DL, and Data Science
  - OMB 6.0
  - Applications: Best Practices

# Overview of the MVAPICH2 Project

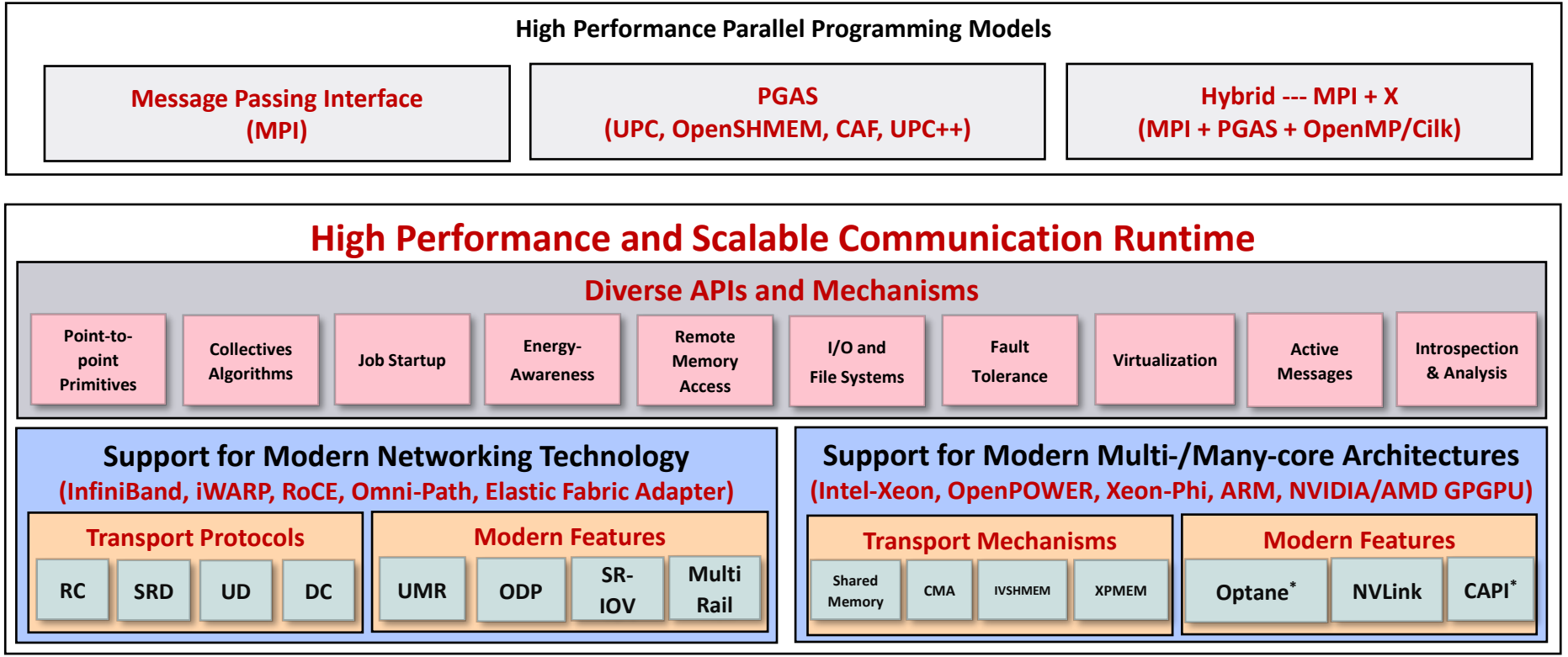
- High Performance open-source MPI Library
- Support for multiple interconnects
  - InfiniBand, Omni-Path, Ethernet/iWARP, RDMA over Converged Ethernet (RoCE), AWS EFA, Rockport Networks, and Slingshot10/11, Broadcom, Cornelis Networks OPX
- Support for multiple platforms
  - x86, OpenPOWER, ARM, Xeon-Phi, GPGPUs (NVIDIA and AMD)
- Started in 2001, first open-source version demonstrated at SC '02
- Supports the latest MPI-3.1 standard
- http://mvapich.cse.ohio-state.edu
- Additional optimized versions for different systems/environments:
  - MVAPICH2-X (Advanced MPI + PGAS), since 2011
  - MVAPICH2-GDR with support for NVIDIA (since 2014) and AMD (since 2020) GPUs
  - MVAPICH2-MIC with support for Intel Xeon-Phi, since 2014
  - MVAPICH2-Virt with virtualization support, since 2015
  - MVAPICH2-EA with support for Energy-Awareness, since 2015
  - MVAPICH2-Azure for Azure HPC IB instances, since 2019
  - MVAPICH2-X-AWS for AWS HPC+EFA instances, since 2019
- Tools:
  - OSU MPI Micro-Benchmarks (OMB), since 2003
  - OSU InfiniBand Network Analysis and Monitoring (INAM), since 2015


- **Used by more than 3,275 organizations in 90 countries**
- **More than 1.63 Million downloads from the OSU site directly**
- Empowering many TOP500 clusters (June '22 ranking)
  - 6th, 10,649,600-core (Sunway TaihuLight) at NSC, China
  - 16th, 448, 448 cores (Frontera) at TACC
  - 30th, 288,288 cores (Lassen) at LLNL
  - 42nd, 570,020 cores (Nurion) in South Korea and many more
- Available with software stacks of many vendors and Linux Distros (RedHat, SuSE, OpenHPC, and Spack)
- Partner in the 16th ranked TACC Frontera system
- **Empowering Top500 systems for more than 20 years**

# MVAPICH2 Release Timeline and Downloads



# Architecture of MVAPICH2 Software Family for HPC and DL/ML




* Upcoming

# Production Quality Software Design, Development and Release

- Rigorous Q&A procedure before making a release
  - Exhaustive unit testing
  - Various test procedures on diverse range of platforms and interconnects
  - Test 19 different benchmarks and applications including, but not limited to
    - OMB, IMB, MPICH Test Suite, Intel Test Suite, NAS, Scalapak, and SPEC
  - Spend about 18,000 core hours per commit
  - Performance regression and tuning
  - Applications-based evaluation
  - Evaluation on large-scale systems
- All versions (alpha, beta, RC1 and RC2) go through the above testing

# MVAPICH2 Software Family

| Requirements | Library |
|---|---|
| MPI with Support for InfiniBand, Omni-Path, Ethernet/iWARP and, RoCE (v1/v2) | MVAPICH2 |
| Optimized Support for Microsoft Azure Platform with InfiniBand | MVAPICH2-Azure |
| Advanced MPI features/support (UMR, ODP, DC, Core-Direct, SHArP, XPMEM), OSU INAM (InfiniBand Network Monitoring and Analysis), | MVAPICH2-X |
| Advanced MPI features (SRD and XPMEM) with support for Amazon Elastic Fabric Adapter (EFA) | MVAPICH2-X-AWS |
| Optimized MPI for clusters with NVIDIA GPUs and for GPU-enabled Deep Learning Applications | MVAPICH2-GDR |
| Energy-aware MPI with Support for InfiniBand, Omni-Path, Ethernet/iWARP and, RoCE (v1/v2) | MVAPICH2-EA |
| MPI Energy Monitoring Tool | OEMT |
| InfiniBand Network Analysis and Monitoring | OSU INAM |
| Microbenchmarks for Measuring MPI and PGAS Performance | OMB |

# Outline

- Brief Overview of the MVAPICH2 Project
- **Features of Recent Releases**
  - **MVAPICH2 2.3.7**
  - MVAPICH2 3.0a
  - MVAPICH2-J
  - MVAPICH2-X-AWS 2.3.7 and Cloud Deployments
  - MVAPICH2-GDR 2.3.7 and Support for ML, DL, and Data Science
  - OMB 6.0
  - Applications: Best Practices
- Upcoming Features
  - Big Data (Spark) over MVAPICH2
  - MVAPICH2-Advanced
  - Conversational AI Interface (CAI)

# MVAPICH2 2.3.7

- **Released on 03/02/2022**
- Major Features and Enhancements
  - Added support for systems with Rockport's switchless networks
    - Added automatic architecture detection
    - Optimized performance for point-to-point operations
  - Added support for the Cray Slingshot 10 interconnect
  - Enhanced support for blocking collective offload using Mellanox SHARP
    - Scatter and Scatterv
  - Enhanced support for non-blocking collective offload using Mellanox SHARP
    - Iallreduce, Ibarrier, Ibcast, and Ireduce
  - Enhanced collective tuning for several systems
  - Add support for GCC compiler v11
  - Add support for Intel IFX compiler
  - Update hwloc v1 code to v1.11.14 & hwloc v2 code to v2.4.2

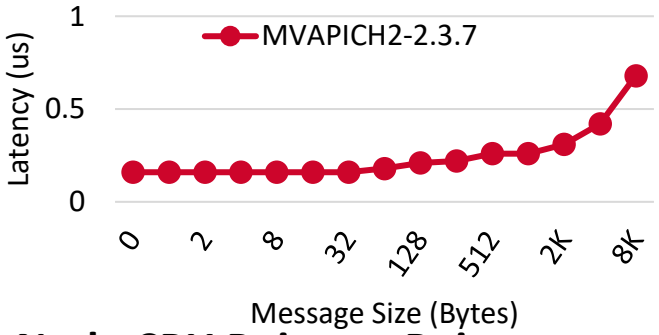
# Highlights of MVAPICH2 2.3.7-GA Release

- Support for highly-efficient inter-node and intra-node communication
- Collective offload using Mellanox's SHARP support
- Performance Engineering with MPI_T
- Support for Newer Adapters
  - Broadcom
  - Rockport Networks
  - Slingshot 10

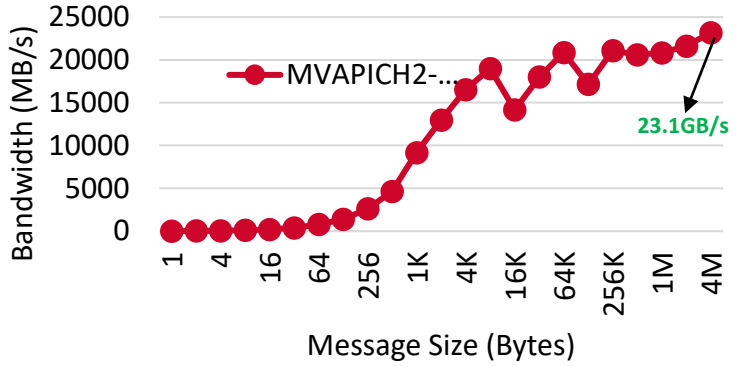
# AMD Milan + HDR 200

## Intra-Node CPU Point-to-Point

### Latency

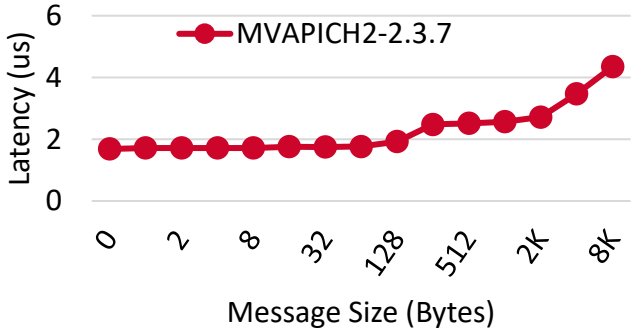



### Bandwidth

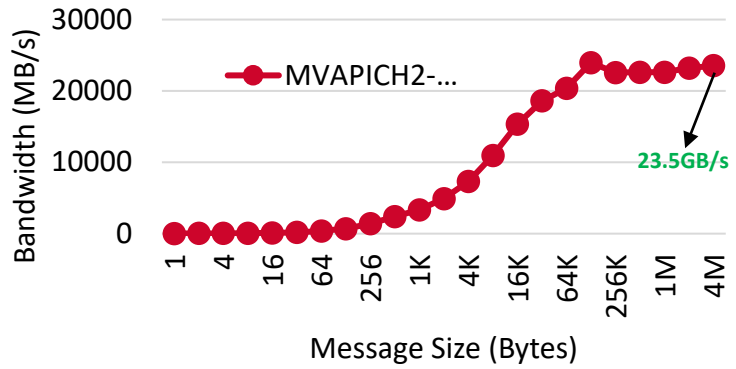



## Inter-Node CPU Point-to-Point

### Latency

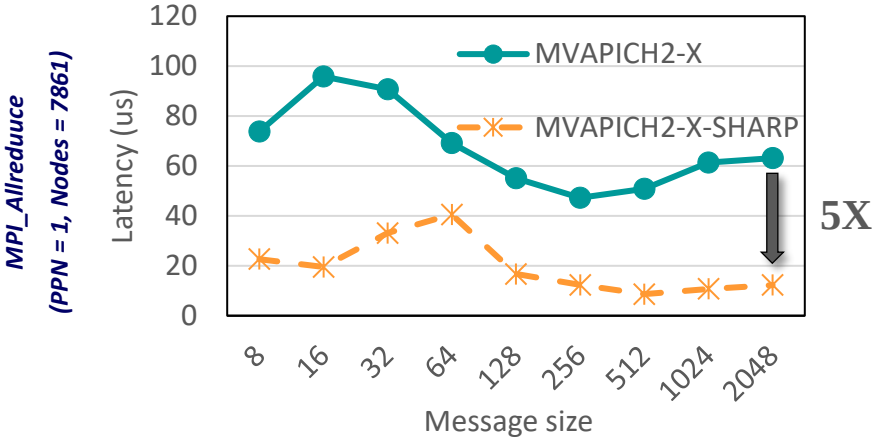


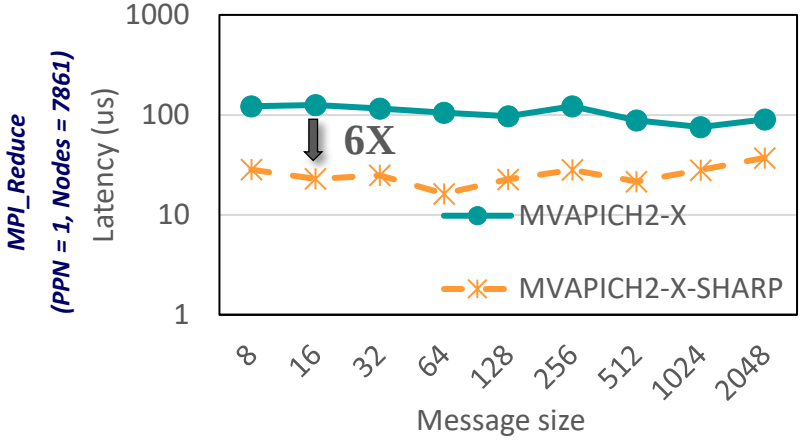

### Bandwidth

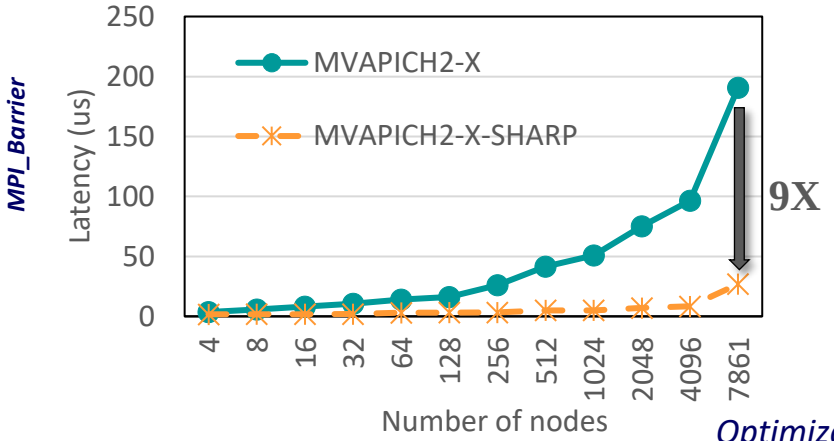



AMD EPYC 7V73X 64-Core Processor, Mellanox ConnectX-6 HDR HCA

# Performance of Collectives with SHARP on TACC Frontera



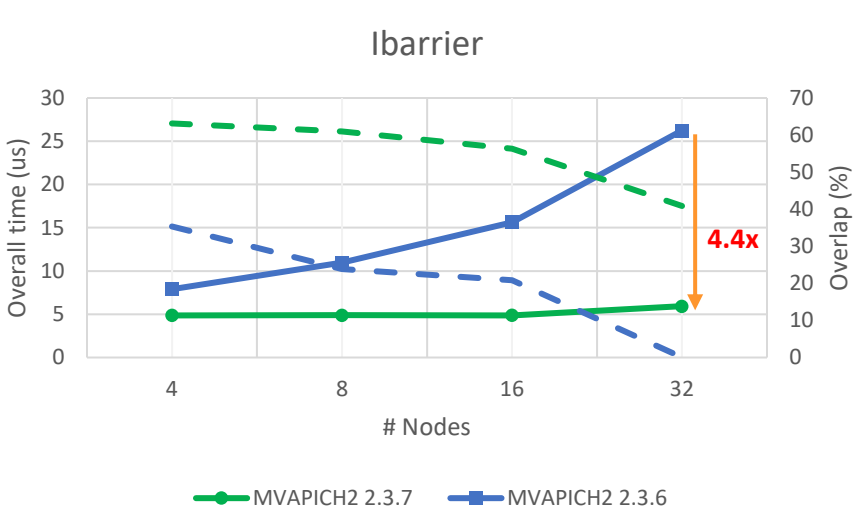

**Optimized SHARP designs in MVAPICH2-X**

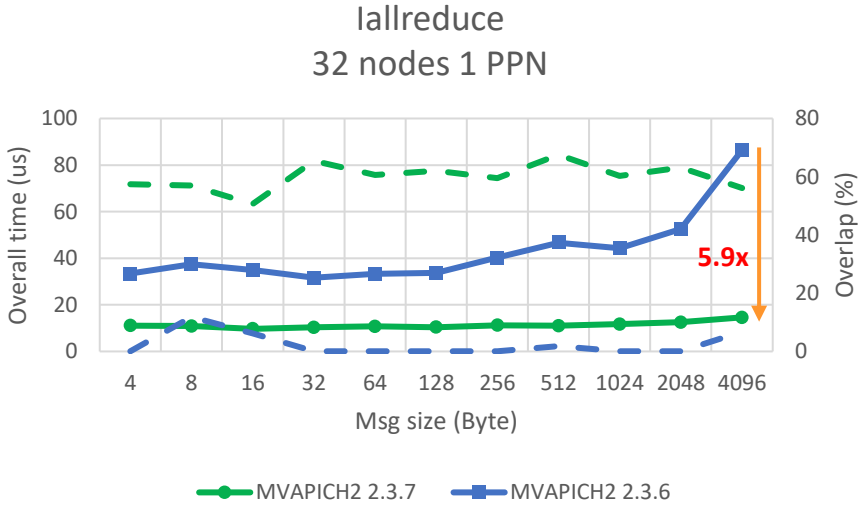
***Up to 9X*** performance improvement with SHARP over MVAPICH2-X default for 1ppn MPI_Barrier, ***6X*** for 1ppn MPI_Reduce and ***5X*** for 1ppn MPI_Allreduce

**B. Ramesh , K. Suresh , N. Sarkauskas , M. Bayatpour , J. Hashmi , H. Subramoni , and D. K. Panda, Scalable MPI Collectives using SHARP: Large Scale Performance Evaluation on the TACC Frontera System, ExaMPI2020 - Workshop on Exascale MPI 2020, Nov 2020.**

*Optimized Runtime Parameters: MV2_ENABLE_SHARP = 1*

# Non-blocking Collectives Support with In-Network Computing

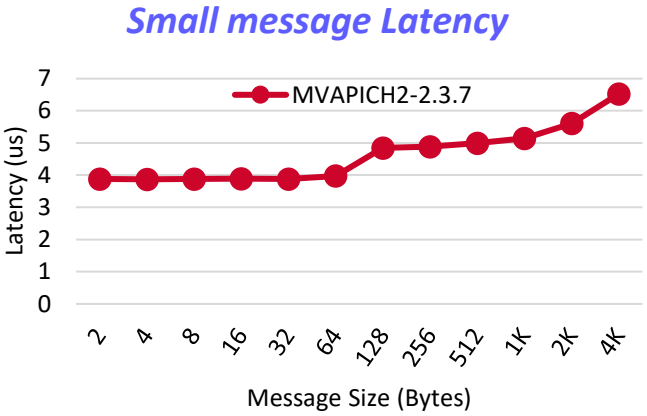


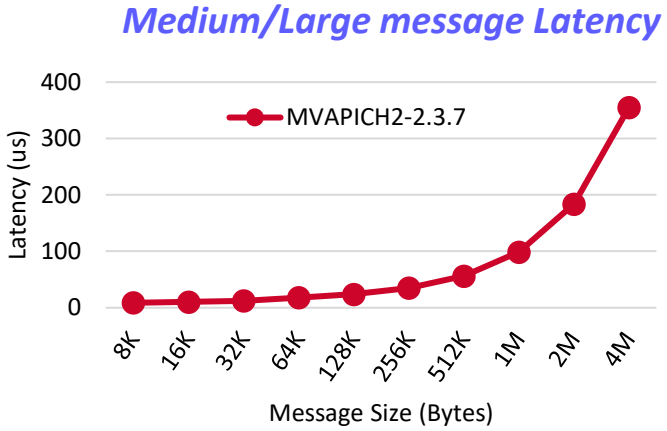




- With SHARP:
  - Flat scaling in terms of overall time
  - High overlap between computation and communication

*Platform: Dual-socket Intel(R) Xeon(R) Platinum 8280 CPU @ 2.70GHz nodes equipped with Mellanox InfiniBand, HDR-100 Interconnect*

# MPI Level Latency on Broadcom RoCE

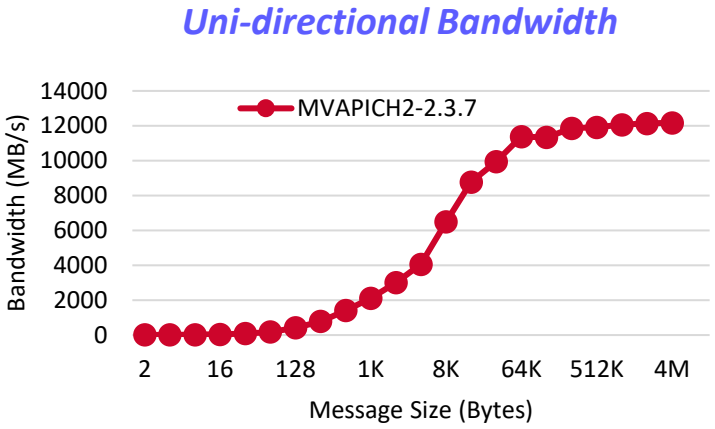


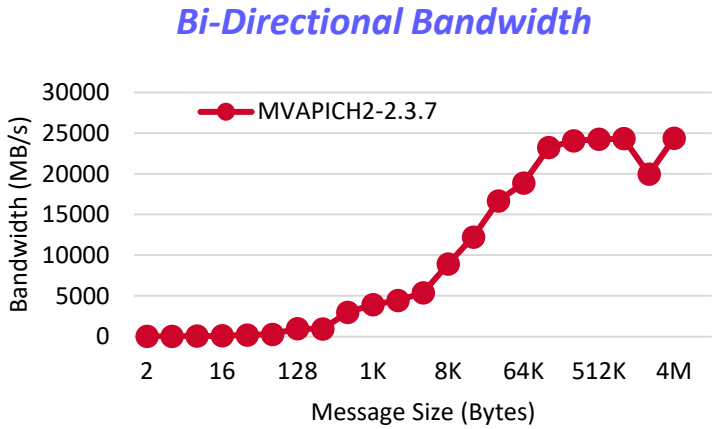




- 3.88us inter-node point-to-point latency for small messages

Interconnect : InfiniBand EDR 100Gbps with Broadcom NetXtreme RoCE HCAs
Library : MVAPICH2 3.0a
CPU : 2 GHz AMD EPYC 7662 64-Core Processor

# MPI Level Latency on Broadcom RoCE



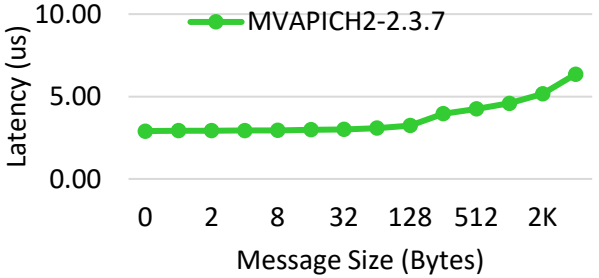




- **12,171 MB/s** uni-directional peak bandwidth
- **24,394 MB/s** bi-directional peak bandwidth

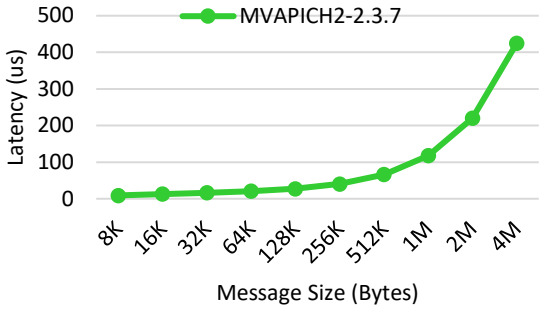
Interconnect : InfiniBand EDR 100Gbps with Broadcom NetXtreme RoCE HCAs
Library : MVAPICH2 3.0a
CPU : 2 GHz AMD EPYC 7662 64-Core Processor

# Inter-node point-to-point Latency and Bandwidth (Rockport Networks)

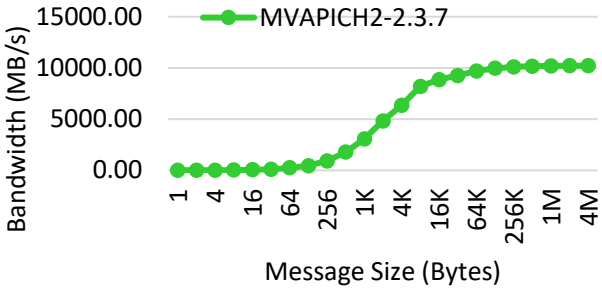
### *Small message Latency*



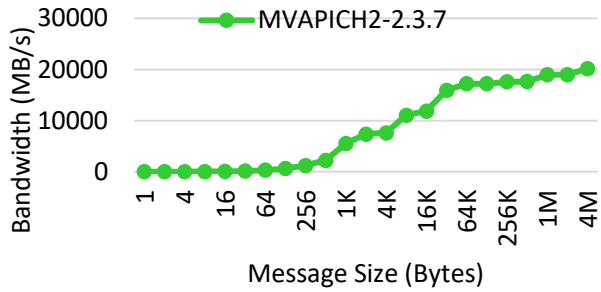

### *Medium/Large message Latency*




### *Uni-directional Bandwidth*




### *Bi-Directional Bandwidth*




- Use multiple QPs to deliver performance

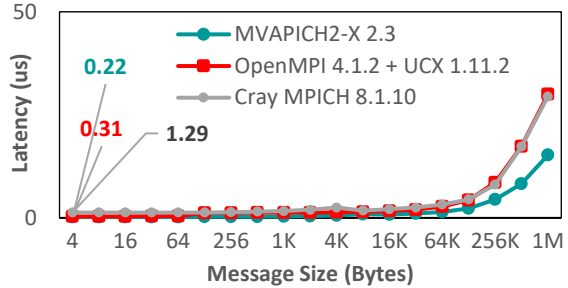
MVAPICH2 delivers

- 3.0 microsec latency for small messages
- 10,229 MB/sec peak unidirectional bandwidth
- 20,165 MB/Sec peak bi-directional bandwidth
- **Available in the MVAPICH2 2.3.7 release**

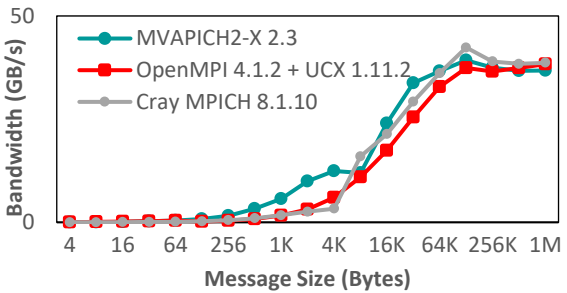
# MVAPICH2 on Slingshot 10 - CPU

## Point-to-Point – Intra-Node

### Latency:

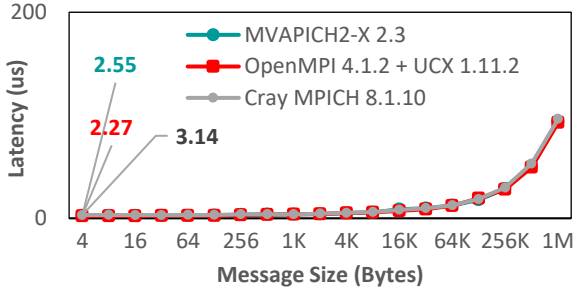



### Bandwidth:

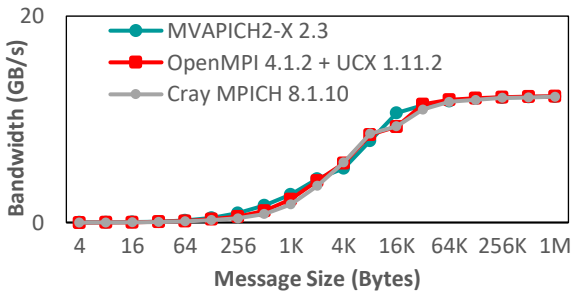



## Point-to-Point – Inter-Node

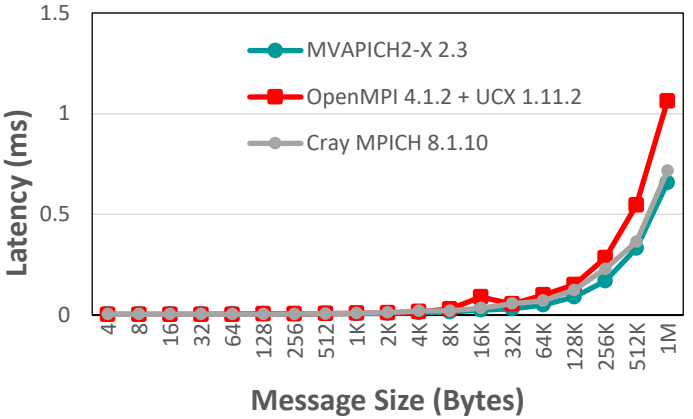
### Latency:



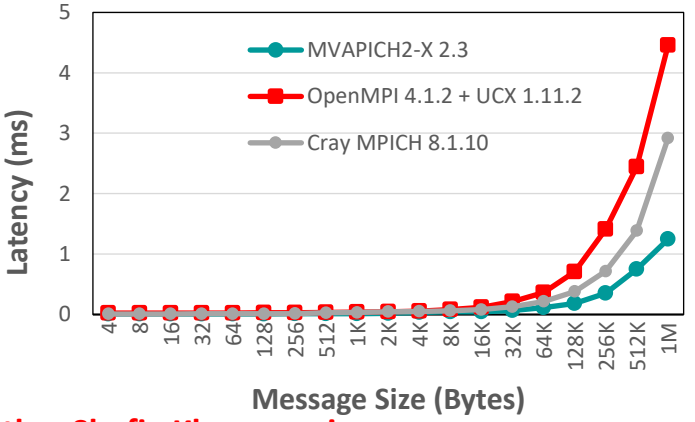

### Bandwidth:




## MPI_Bcast (4 Nodes, 64PPN)




## MPI_Allreduce (4 Nodes, 64PPN)




## AMD Epyc Rome CPUs

More details in today's talk by Kawthar Shafie Khorassani

# Outline

- Brief Overview of the MVAPICH2 Project
- **Features of Recent Releases**
  - MVAPICH2 2.3.7
  - **MVAPICH2 3.0a**
  - MVAPICH2-J
  - MVAPICH2-X-AWS 2.3.7 and Cloud Deployments
  - MVAPICH2-GDR 2.3.7 and Support for ML, DL, and Data Science
  - OMB 6.0
  - Applications: Best Practices

# MVAPICH2-3.0a

- Released on 08/15/2022
- Based on MPICH 3.4.3
- Added support for the ch4:ucx and ch4:ofi devices
- Support for MVAPICH2 enhanced collectives over OFI and UCX
- Added support for the Cray Slingshot 11 interconnect over OFI
  - Supports Cray Slingshot 11 network adapters
- Added support for the Cornelis OPX library over OFI
  - Supports Intel Omni-Path adapters
- Added support for the Intel PSM3 library over OFI
  - Supports Intel Columbiaville network adapters
- Added support for IB verbs over UCX
  - Supports IB and RoCE network adapters

# MVAPICH-Plus

- Released on 11/11/2022
- Based on MVAPICH 3.0
- Advanced MPI with unified MVAPICH2-GDR and MVAPICH2-X features
- Support for both OFI and UCX
- Support for NVIDIA and AMD GPUs
- Support for all HPC interconnects
  - InfiniBand, Omni-Path, ROCE, Slingshot
- Optimized designs for HPC, DL, ML, Big Data and Data Science applications

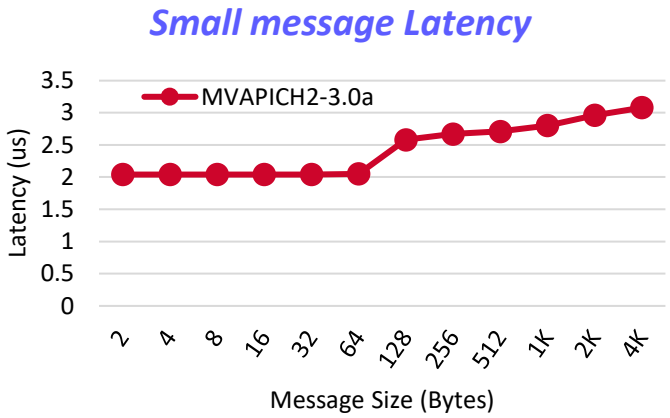
# Highlights of MVAPICH2 3.0a Release

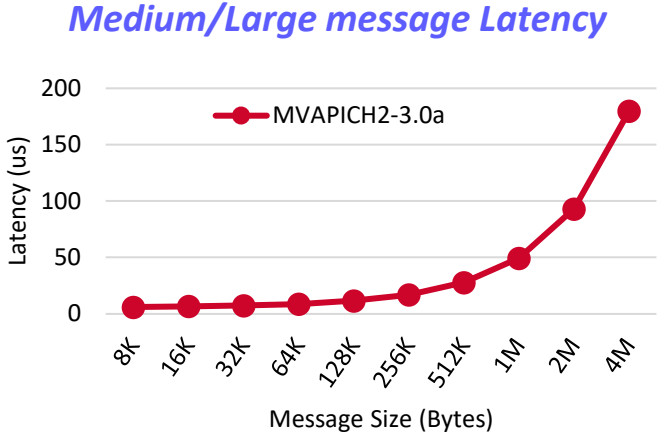
- Support for OFI and UCX
- Support for Newer Networks
  - Slingshot 11
  - OPX over Omni-Path

# Features of OFI and UCX Support

- Support a broad range of interconnects with widely used libraries
  - Configure with `--with-device=ch4:ofi` or `--with-device=ch4:ucx`
- Runtime provider selection via CVARs
  - `MPIR_CVAR_OFI_USE_PROVIDER=<prov>`
- System default, embedded, or custom installation of OFI/UCX
  - Configure with `--with-libfabric=embedded` or `--with-libfabric=<path>`
  - Configure with `--with-ucx=embedded` or `--with-ucx=<path>`
- Enhanced MVAPICH2 collective designs

# MPI Level Latency on Slingshot 11



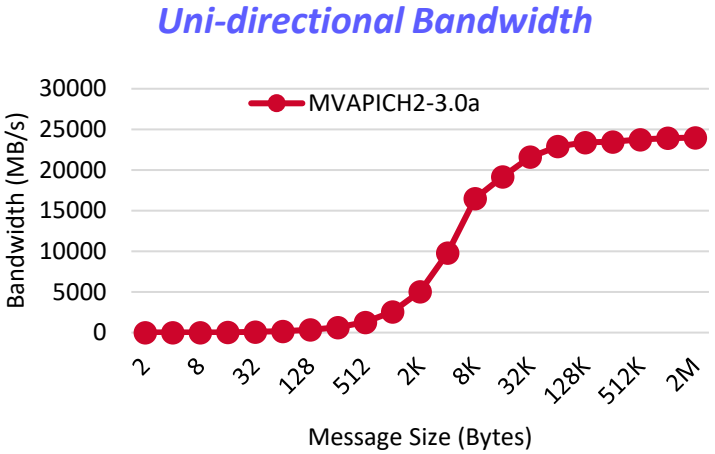

- **2us** inter-node point-to-point latency for small messages

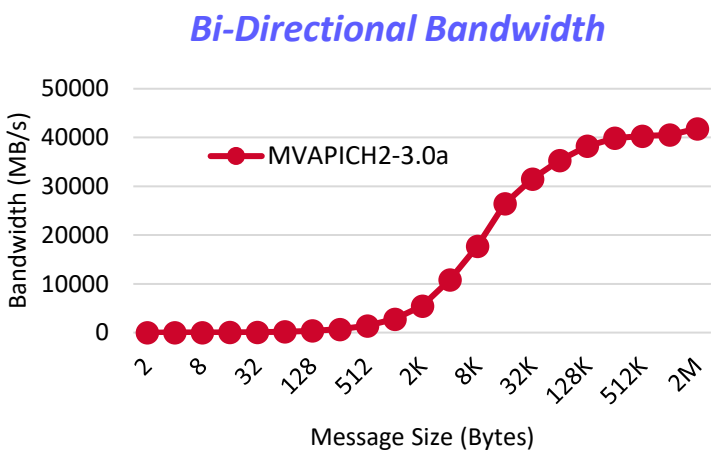
Interconnect : Cray HPE Slingshot 11
Library : MVAPICH2 3.0a
CPU : AMD EPYC 7763 (milan) Processor

**More details in today's talk by**

**Kawthar Shafie Khorassani**

# MPI Level Bandwidth on Slingshot 11

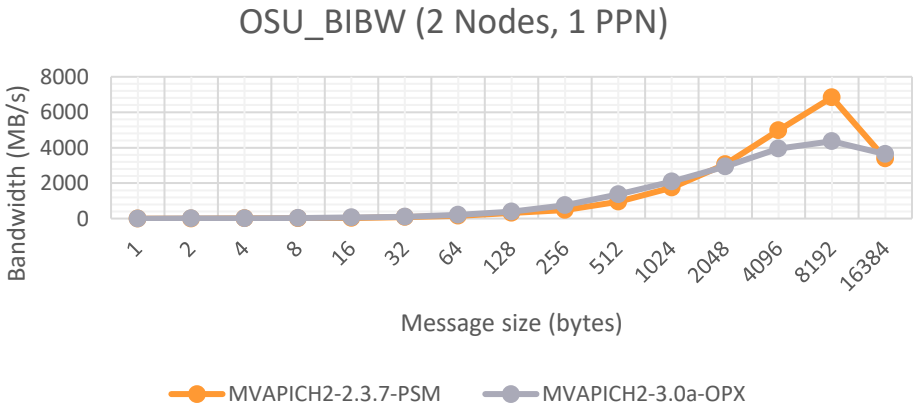


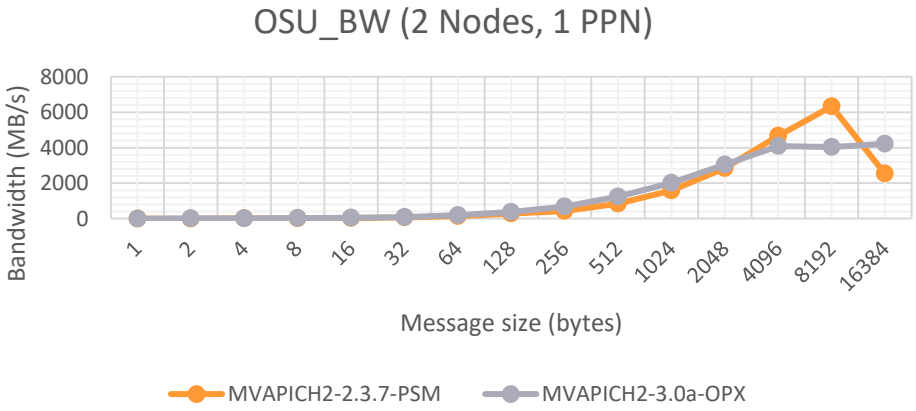



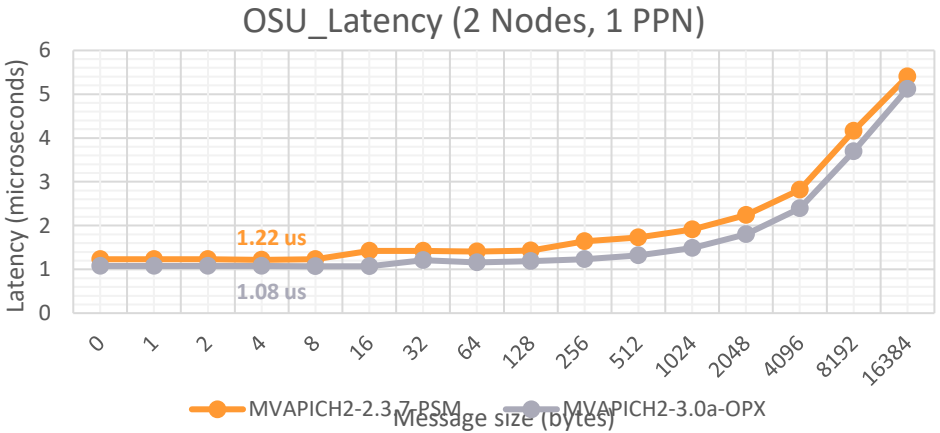

- **23,985 MB/s** uni-directional peak bandwidth
- **42,034 MB/s** bi-directional peak bandwidth

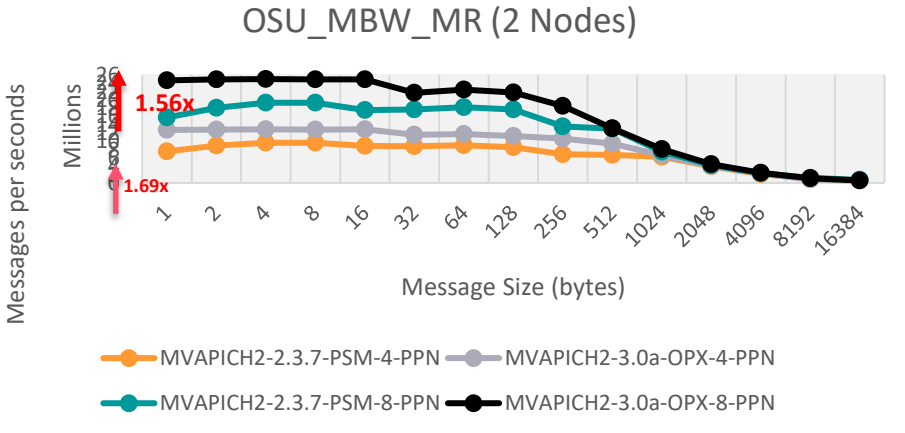
Interconnect : Cray HPE Slingshot 11 (200 Gbps)
Library : MVAPICH2 3.0a
CPU : AMD EPYC 7763 (milan) Processor

# MVAPICH2-3.0a+OPX vs MVAPICH2-2.3.7+PSM2 (Early Performance Results)



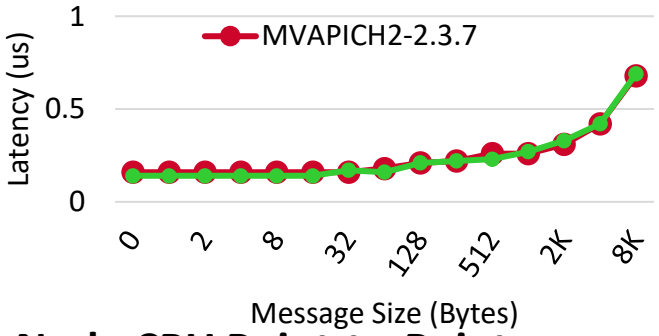




System: Intel Xeon Bronze (Skylake) 3106 CPU @ 1.70GHz (4 nodes, 16 cores/node, 8 x 2 sockets) with Omni-Path 100Gbps

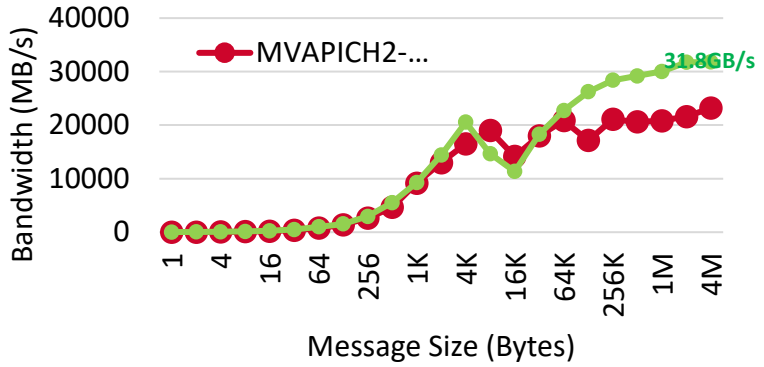
# AMD Milan + HDR 200

## Intra-Node CPU Point-to-Point

### Latency

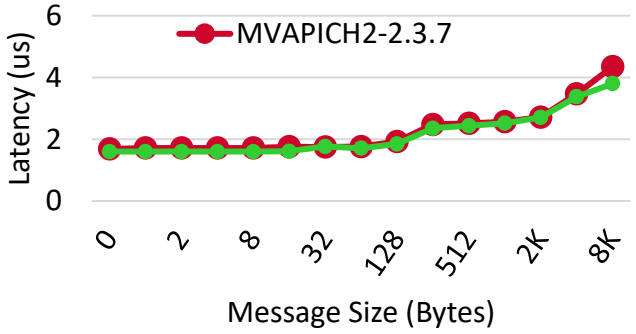



### Bandwidth

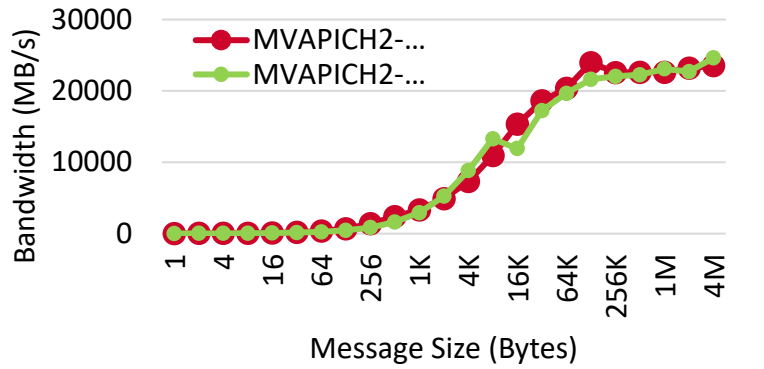



## Inter-Node CPU Point-to-Point

### Latency




### Bandwidth




AMD EPYC 7V73X 64-Core Processor, Mellanox ConnectX-6 HDR HCA

# Outline

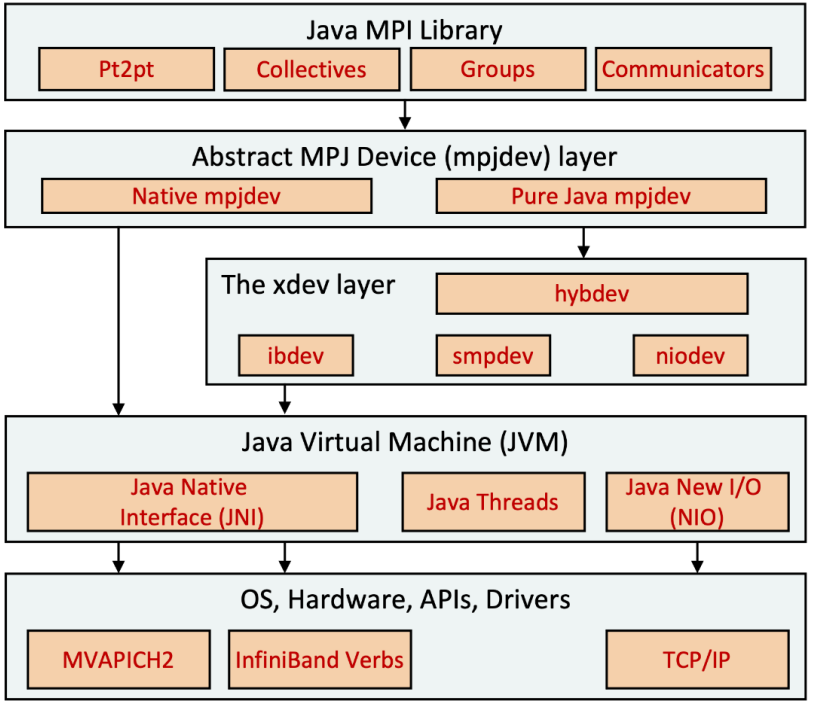
- Brief Overview of the MVAPICH2 Project
- **Features of Recent Releases**
  - MVAPICH2 2.3.7
  - MVAPICH2 3.0a
  - **MVAPICH2-J**
  - MVAPICH2-X-AWS 2.3.7 and Cloud Deployments
  - MVAPICH2-GDR 2.3.7 and Support for ML, DL, and Data Science
  - OMB 6.0
  - Applications: Best Practices

# MVAPICH2-J

- Released on 07/08/2022
- Features and Enhancements
  - Provides Java bindings to the MVAPICH2 family of libraries
  - Support for communication of basic Java datatypes and Java new I/O (NIO) package direct ByteBuffers
  - Support for blocking and non-blocking point-point communication protocols
  - Support for blocking collective and strided collective communication protocols
  - Support for Dynamic Process Management (DPM) functionality
  - Utilizes a pool of direct Java NIO ByteBuffers used in communication of basic datatype Java arrays
  - Relies on the explicit memory management buffering layer from the MPJ Express software
  - Support for all high-speed interconnects that MVAPICH2 supports including InfiniBand, Internet Wide-area RDMA Protocol (iWARP), RDMA over Converged Ethernet (RoCE), Intel's Performance Scaled Messaging (PSM), Omni-Path, etc.

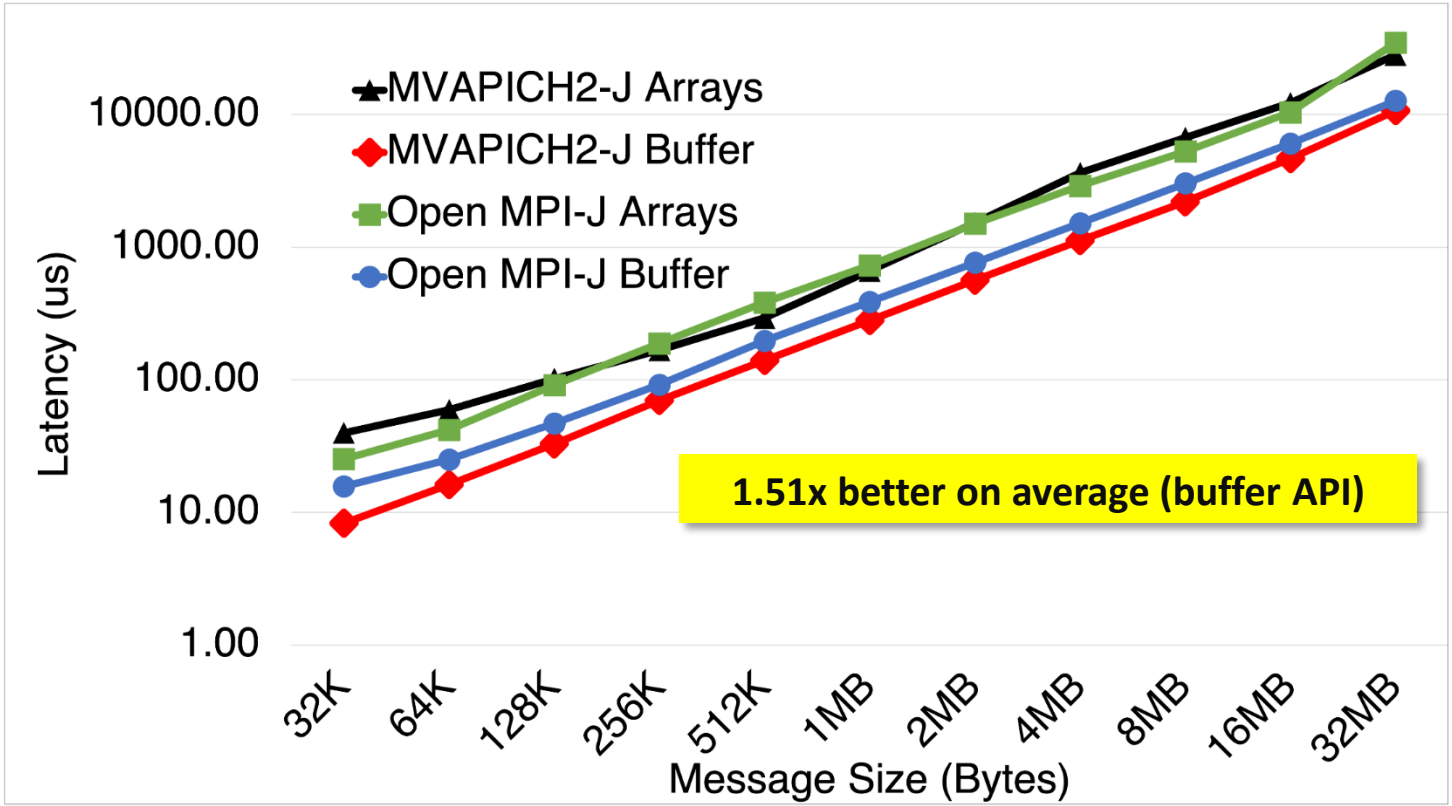
# MVAPICH2-J: Java Bindings to MVAPICH2

- We have recently added Java bindings to the MVAPICH2 library:
  - Allows writing HPC applications in the Java programming language
- The library currently implements a subset of the MPI API:
  - Our bindings follow the same API as Open MPI Java bindings
- MVAPICH2-J currently supports:
  - blocking/non-blocking point-to-point functions
  - blocking collective functions
  - blocking vectored collective functions
- Motivation:
  - Enhance communication infrastructure of BigData frameworks, written in Scala/Java, using MPI
- MVAPICH2-J 2.3.7 is recently released:
  - User guide: https://mvapich.cse.ohio-state.edu/userguide/mv2j/




K. Al Attar, A. Shafi, H. Subramoni, D. Panda, Towards Java-based HPC using the MVAPICH2 Library: Early Experiences, HIPS '22 (IPDPSW), May 2022.

# Java Binding in MVAPICH2: Preliminary Results Bcast Performance (8 processes)



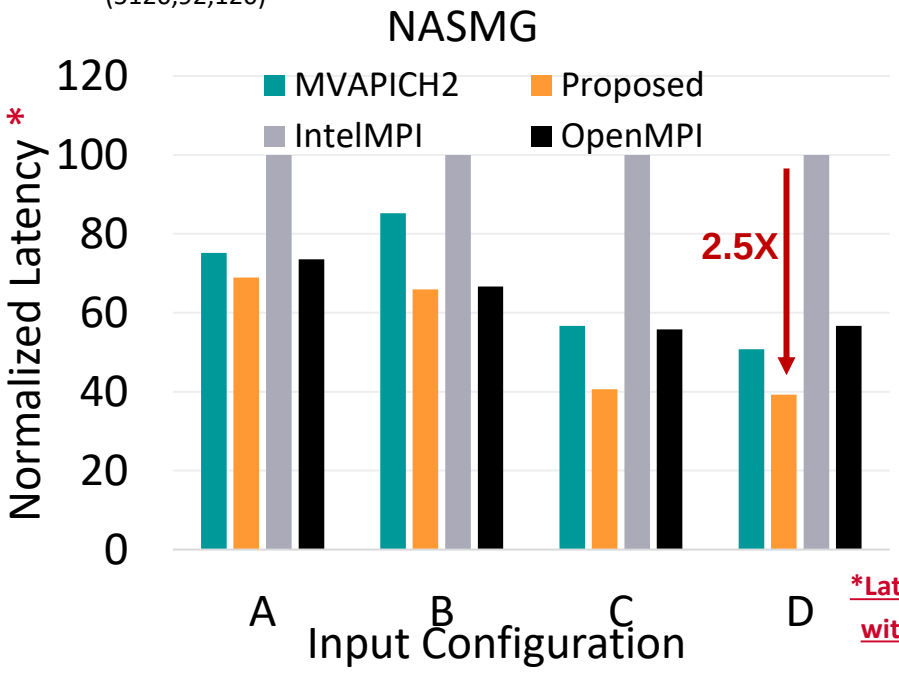

More details were in yesterday's talk by Aamir Shafi & Nawaras Alnassan

# MVAPICH2 Software Family

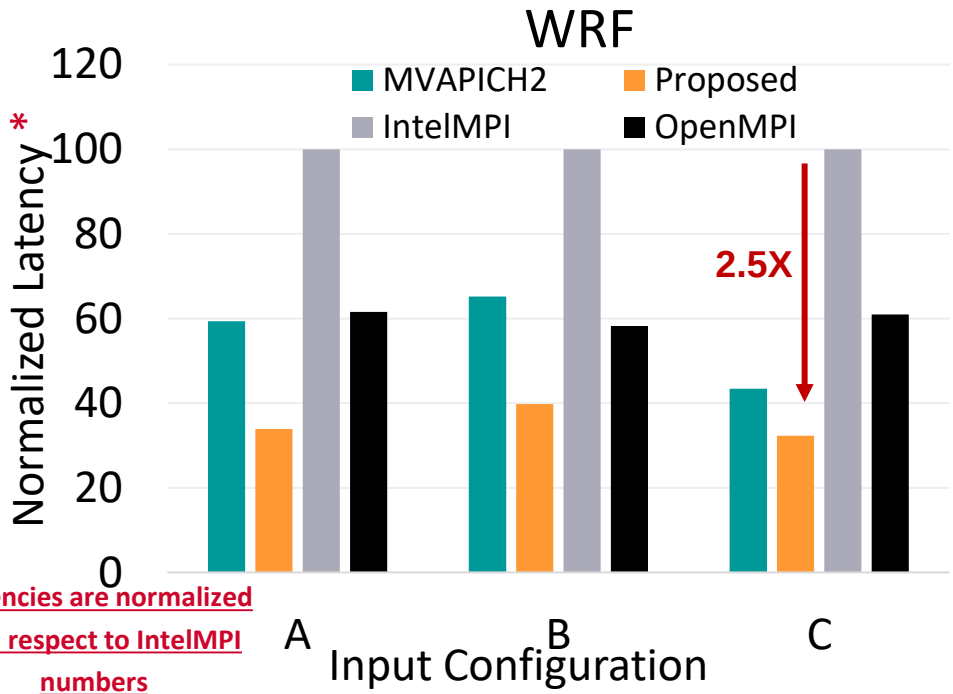
| Requirements | Library |
|---|---|
| MPI with Support for InfiniBand, Omni-Path, Ethernet/iWARP and, RoCE (v1/v2) | MVAPICH2 |
| Optimized Support for Microsoft Azure Platform with InfiniBand | MVAPICH2-Azure |
| Advanced MPI features/support (UMR, ODP, DC, Core-Direct, SHArP, XPMEM), OSU INAM (InfiniBand Network Monitoring and Analysis) | MVAPICH2-X |
| Advanced MPI features (SRD and XPMEM) with support for Amazon Elastic Fabric Adapter (EFA) | MVAPICH2-X-AWS |
| Optimized MPI for clusters with NVIDIA GPUs and for GPU-enabled Deep Learning Applications | MVAPICH2-GDR |
| Energy-aware MPI with Support for InfiniBand, Omni-Path, Ethernet/iWARP and, RoCE (v1/v2) | MVAPICH2-EA |
| MPI Energy Monitoring Tool | OEMT |
| InfiniBand Network Analysis and Monitoring | OSU INAM |
| Microbenchmarks for Measuring MPI and PGAS Performance | OMB |

# Performance of DDTbench with Optimized Derived Datatype Support

- NASMG: Block length is 8 bytes for X-direction and 256 bytes to 5KB in the Y-direction
- 28% improvement over MVAPICH2 and 2.5X over IntelMPI
- Inputs : A = (256,32,32) B = (512,66,66) C = (2048,66,120) D = (5120,92,120)

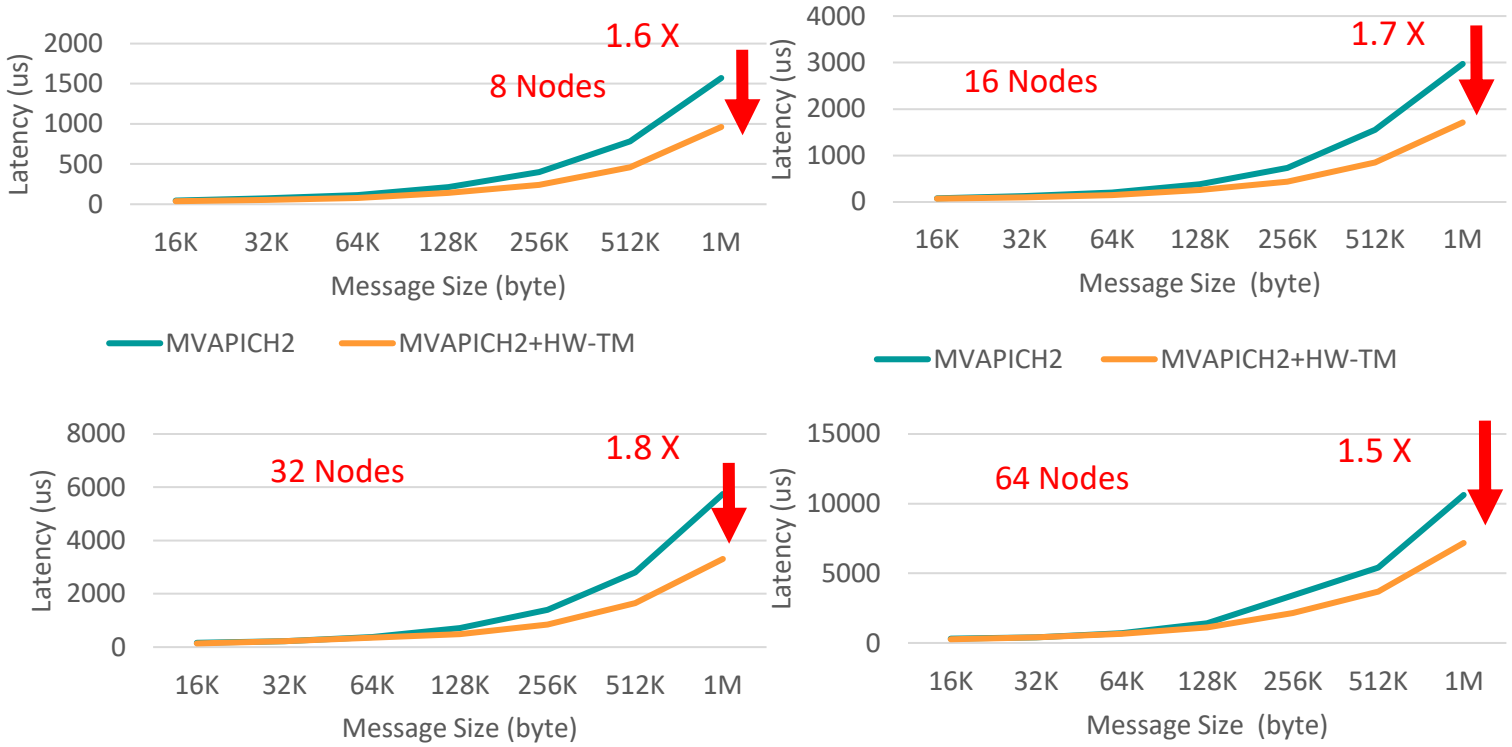



- WRF: The datatypes used in WRF are struct of vectors for both X and Y direction
- We see improvements up to 1.75X compared to MVAPICH2 and up to 2.5X improvements over IntelMPI
- Inputs : A = (4,4018,8,4010) B = (4,2060,8,2056) C = (4,6012,8,6008)




*Latencies are normalized with respect to IntelMPI numbers

# Performance of MPI_Ialltoall using HW Tag Matching




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

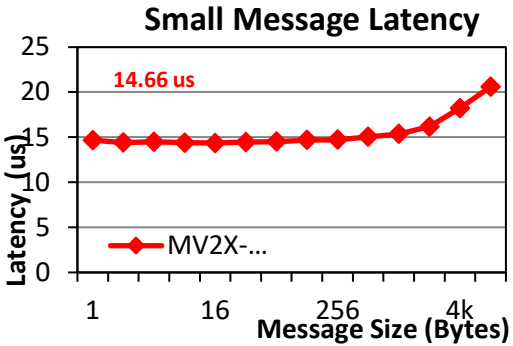
# Outline

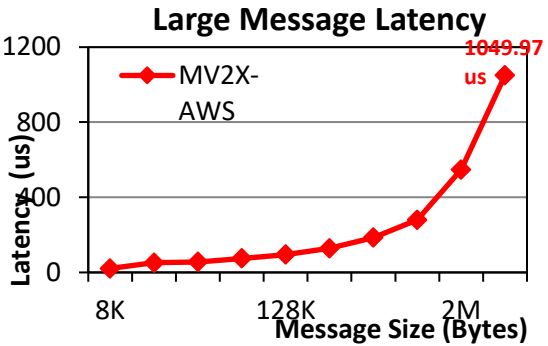
- Brief Overview of the MVAPICH2 Project
- **Features of Recent Releases**
  - MVAPICH2 2.3.7
  - MVAPICH2 3.0a
  - MVAPICH2-J
  - **MVAPICH2-X-AWS 2.3.7 and Cloud Deployments**
  - MVAPICH2-GDR 2.3.7 and Support for ML, DL, and Data Science
  - OMB 6.0
  - Applications: Best Practices

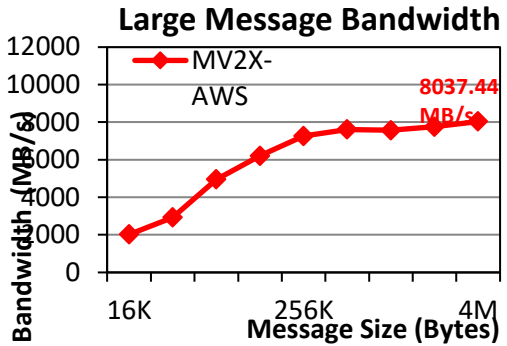
# MVAPICH2-X-AWS 2.3.7

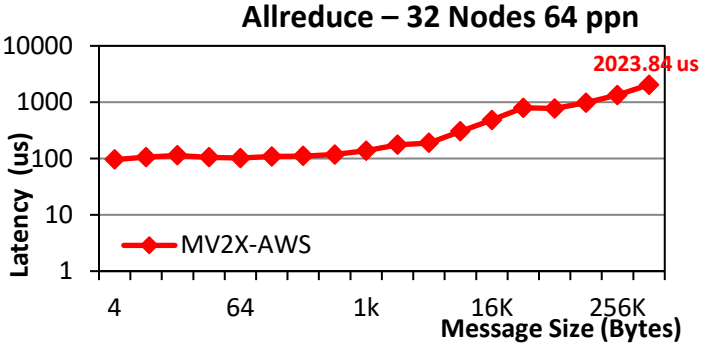
- Released on 08/21/2022
- Features and Enhancements
  - Based on MVAPICH2-X
  - Improved inter-node latency & bandwidth performance
  - Add initial support for AWS hpc6a/c6a instances with 3rd generation AMD EPYC processors
  - Add support & performance optimization for AWS c6g/c7g with Amazon Graviton 2/3 ARM processors
  - Add initial support to rdma_read feature for AWS p4d instance type
  - Support for currently available basic OS types on AWS EC2 including: Amazon Linux 1/2, CentOS 7, Ubuntu 18.04/20.04

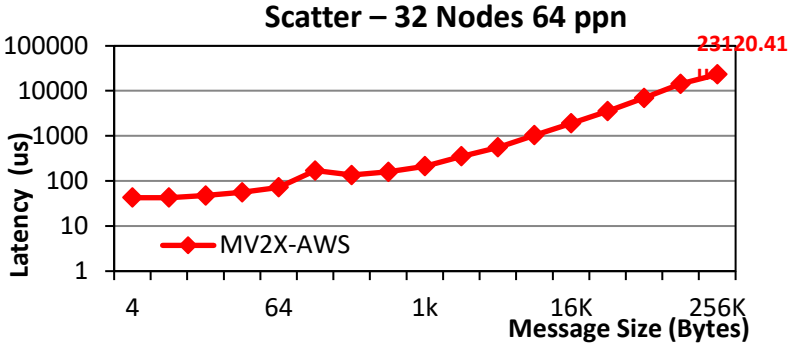
# MVAPICH2-X-AWS Performance on AWS Arm (c6gn) Instances



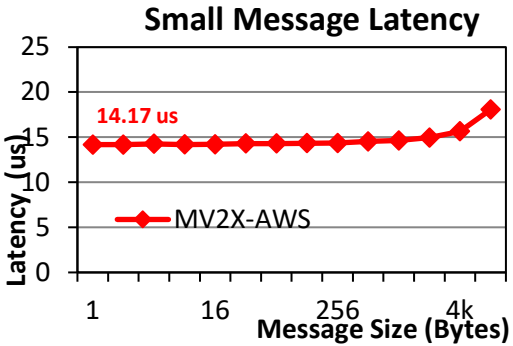



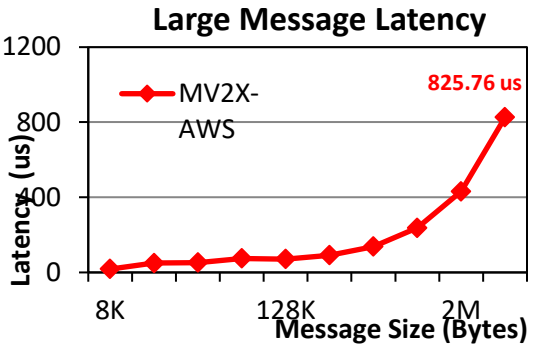



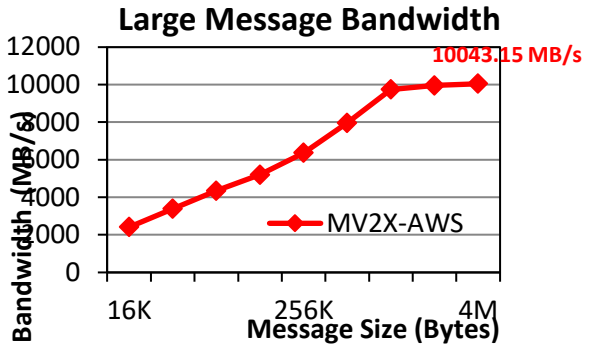



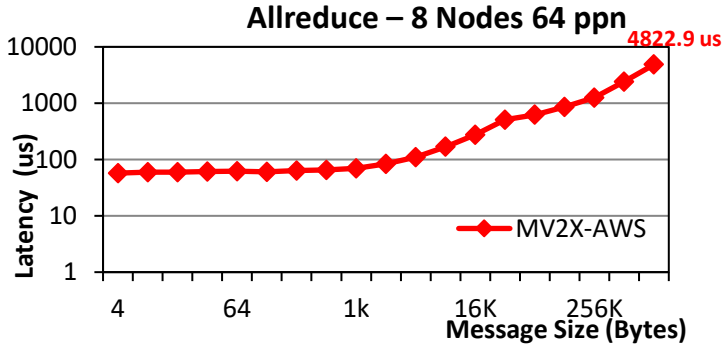

**More details in the poster by Shulei Xu**

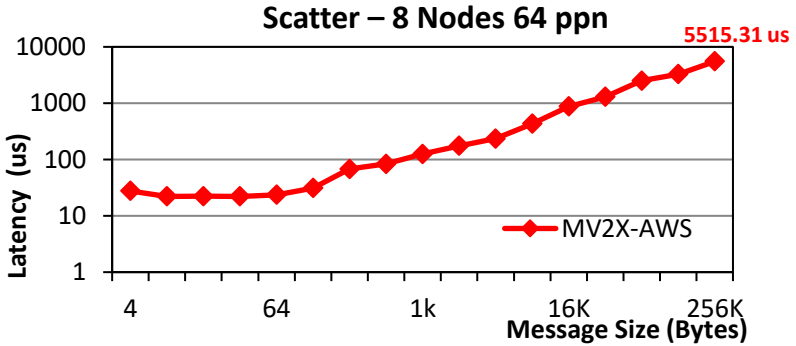
# MVAPICH2-X-AWS Performance on AWS x86 Instances w/ AMD EPYC Gen3



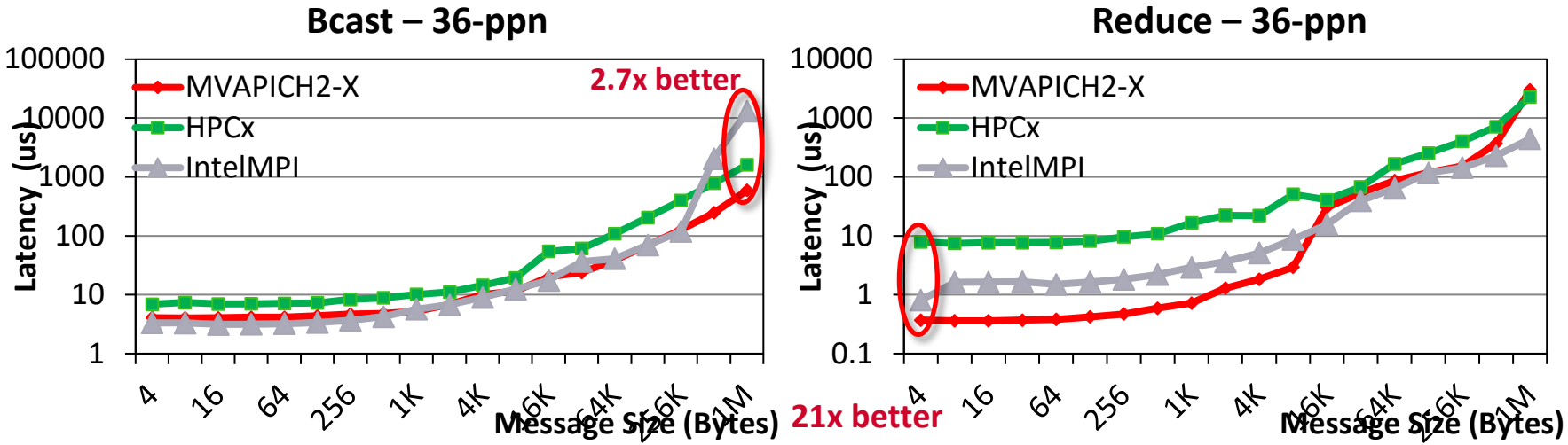



# MVAPICH2-X Performance on OCI HPC System

- Collective performance evaluation on 8 BM.HPC2 instances



# Outline

- Brief Overview of the MVAPICH2 Project
- **Features of Recent Releases**
  - MVAPICH2 2.3.7
  - MVAPICH2 3.0a
  - MVAPICH2-J
  - MVAPICH2-X-AWS 2.3.7 and Cloud Deployments
  - **MVAPICH2-GDR 2.3.7 and Support for ML, DL, and Data Science**
  - OMB 6.0
  - Applications: Best Practices

# MVAPICH2 Software Family

| Requirements | Library |
|---|---|
| MPI with Support for InfiniBand, Omni-Path, Ethernet/iWARP and, RoCE (v1/v2) | MVAPICH2 |
| Optimized Support for Microsoft Azure Platform with InfiniBand | MVAPICH2-Azure |
| Advanced MPI features/support (UMR, ODP, DC, Core-Direct, SHArP, XPMEM), OSU INAM (InfiniBand Network Monitoring and Analysis) | MVAPICH2-X |
| Advanced MPI features (SRD and XPMEM) with support for Amazon Elastic Fabric Adapter (EFA) | MVAPICH2-X-AWS |
| Optimized MPI for clusters with NVIDIA GPUs and for GPU-enabled Deep Learning Applications | MVAPICH2-GDR |
| Energy-aware MPI with Support for InfiniBand, Omni-Path, Ethernet/iWARP and, RoCE (v1/v2) | MVAPICH2-EA |
| MPI Energy Monitoring Tool | OEMT |
| InfiniBand Network Analysis and Monitoring | OSU INAM |
| Microbenchmarks for Measuring MPI and PGAS Performance | OMB |

# MVAPICH2-GDR 2.3.7

- Released on 05/27/2022
- Major Features and Enhancements
  - Based on MVAPICH2 2.3.7
  - Enhanced performance for GPU-aware MPI_Alltoall and MPI_Alltoallv
  - Added automatic rebinding of processes to cores based on GPU NUMA domain
    - This is enabled by setting the env MV2_GPU_AUTO_REBIND=1
  - Added NCCL communication substrate for various non-blocking MPI collectives
    - MPI_Iallreduce, MPI_Ireduce, MPI_Iallgather, MPI_Iallgatherv, MPI_Ialltoall, MPI_Ialltoallv, MPI_Iscatter, MPI_Iscatterv, MPI_Igather, MPI_Igatherv, and MPI_Ibcast
  - Enhanced point-to-point and collective tuning for AMD Milan processors with NVIDIA A100 and AMD Mi100 GPUs
  - Enhanced point-to-point and collective tuning for NVIDIA DGX-A100 systems
  - Added support for Cray Slingshot-10 interconnect
  - Added support for 'on-the-fly' compression of point-to-point messages used for GPU-to-GPU communication
    - Applicable to NVIDIA GPUs
  - NCCL communication substrate for various MPI collectives
    - Support for hybrid communication protocols using NCCL-based, CUDA-based, and IB verbs-based primitives
    - MPI_Allreduce, MPI_Reduce, MPI_Allgather, MPI_Allgatherv, MPI_Alltoall, MPI_Alltoallv, MPI_Scatter, MPI_Scatterv, MPI_Gather, MPI_Gatherv, and MPI_Bcast
  - Full support for NVIDIA DGX, NVIDIA DGX-2 V-100, and NVIDIA DGX-2 A-100 systems
  - Enhanced architecture detection, process placement and HCA selection
  - Enhanced intra-node and inter-node point-to-point tuning
  - Enhanced collective tuning
  - Introduced architecture detection, point-to-point tuning and collective tuning for ThetaGPU @ANL
  - Enhanced point-to-point and collective tuning for NVIDIA GPUs on Frontera @TACC, Lassen @LLNL, and Sierra @LLNL
  - Enhanced point-to-point and collective tuning for Mi50 and Mi60 AMD GPUs on Corona @LLNL
  - Added several new MPI_T PVARs
  - Added support for CUDA 11.3
  - Added support for ROCm 4.1
  - Enhanced output for runtime variable MV2_SHOW_ENV_INFO
  - Tested with Horovod and common DL Frameworks
    - TensorFlow, PyTorch, and MXNet
  - Tested with MPI4Dask 0.2
    - MPI4Dask is a custom Dask Distributed package with MPI support
  - Tested with MPI4cuML 0.1
    - MPI4cuML is a custom cuML package with MPI support

# Highlights of some MVAPICH2-GDR Features for HPC and DL

- CUDA-Aware MPI
- Derived Data Type (DDT) Support
- On-the-fly Compression for GPU-GPU Communication
- Optimized Collective Support for DGX-A100
- Support for
  - AMD GPUs
  - Slingshot 10 + AMD GPUs
- High-Performance
  - Deep Learning
  - Machine Learning
  - Data Science with Dask

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

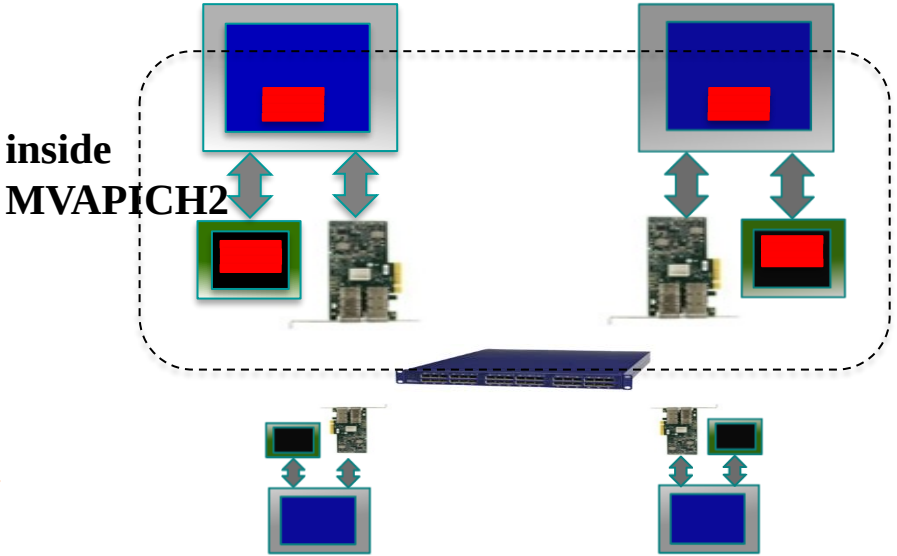
- Standard MPI interfaces used for unified data movement
- Takes advantage of Unified Virtual Addressing (>= CUDA 4.0)
- Overlaps data movement from GPU with RDMA transfers

**At Sender:**

```
MPI_Send(s_devbuf, size, …);
```

**At Receiver:**

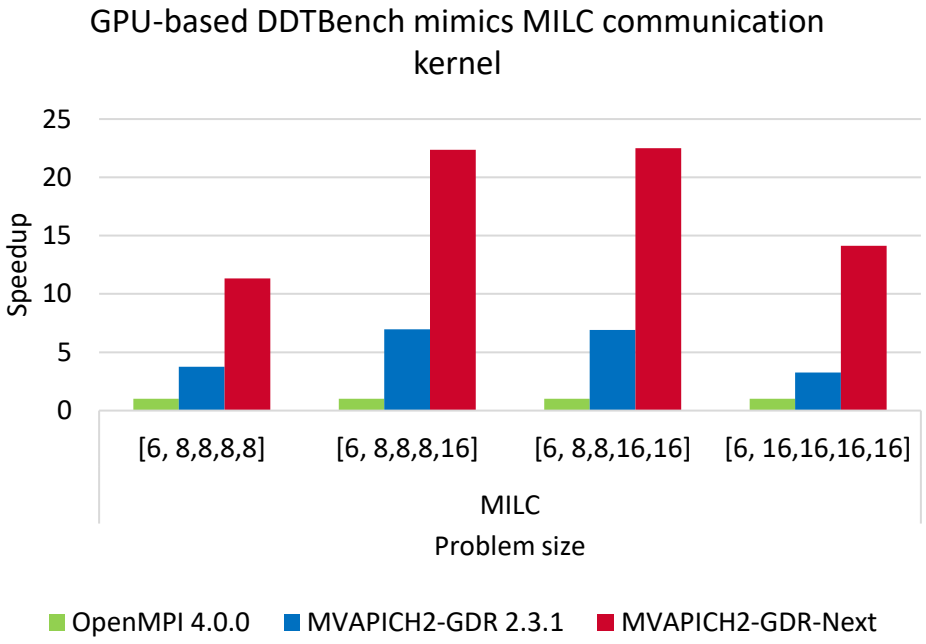
```
MPI_Recv(r_devbuf, size, …);
```




***High Performance and High Productivity***

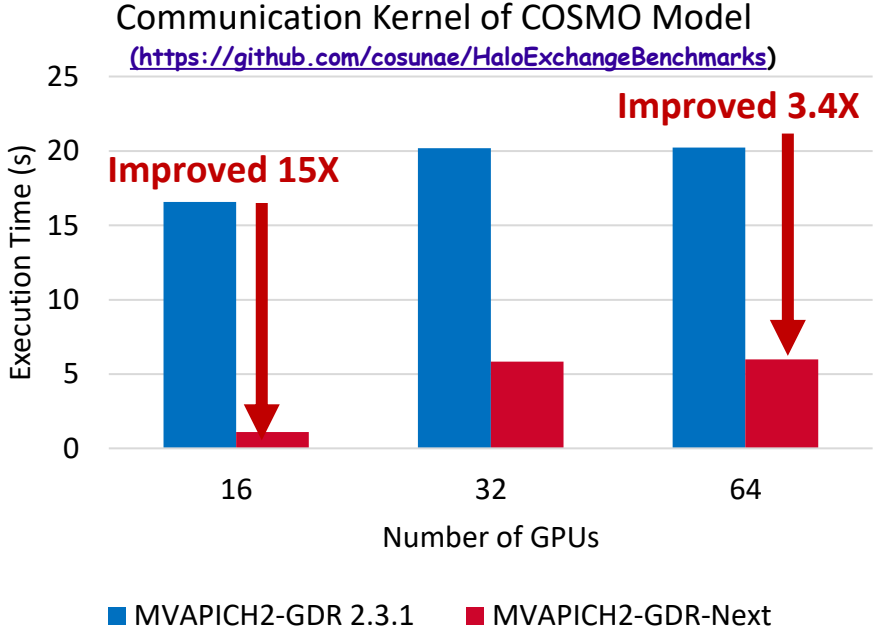
# MVAPICH2-GDR: Enhanced Derived Datatype

- **Kernel-based and GDRCOPY-based one-shot packing for inter-socket and inter-node communication**
- **Zero-copy (packing-free) for GPUs with peer-to-peer direct access over PCIe/NVLink**




*Platform: Nvidia DGX-2 system*

*(NVIDIA Volta GPUs connected with NVSwitch), CUDA 9.2*

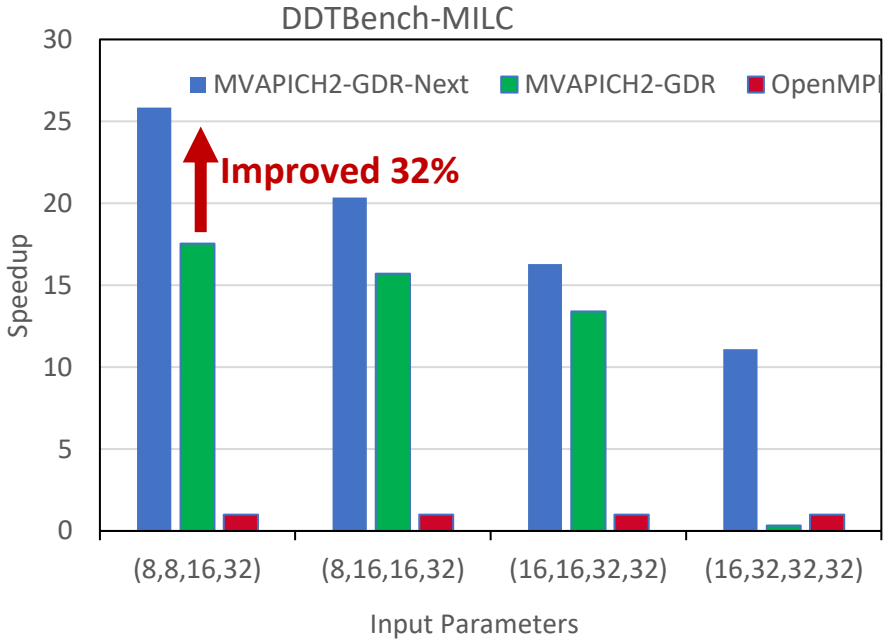


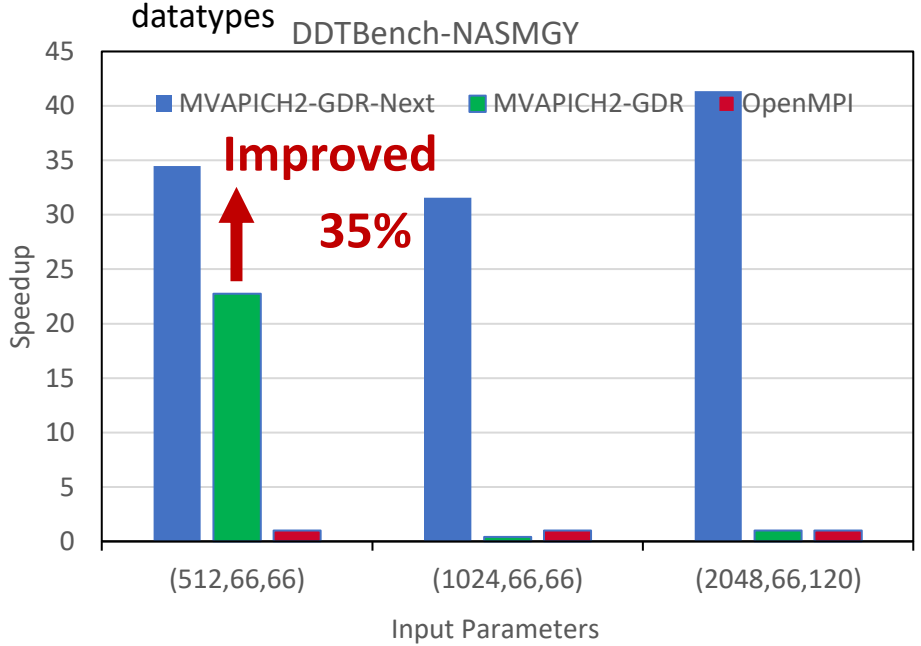

*Platform: Cray CS-Storm*

*(16 NVIDIA Tesla K80 GPUs per node), CUDA 8.0*

# Enhanced DDT Support: HCA Assisted Inter-Node Scheme (UMR)

- **Comparison of UMR based DDT scheme in MVAPICH2-GDR-Next with OpenMPI 4.1.3, MVAPICH2-GDR 2.3.6**
- **1 GPU per Node, 2 Node experiment. Speed-up relative to OpenMPI**
- Uses nested vector datatype for 4D face exchanges.
- 3D face exchanges with vector and nested vector datatypes



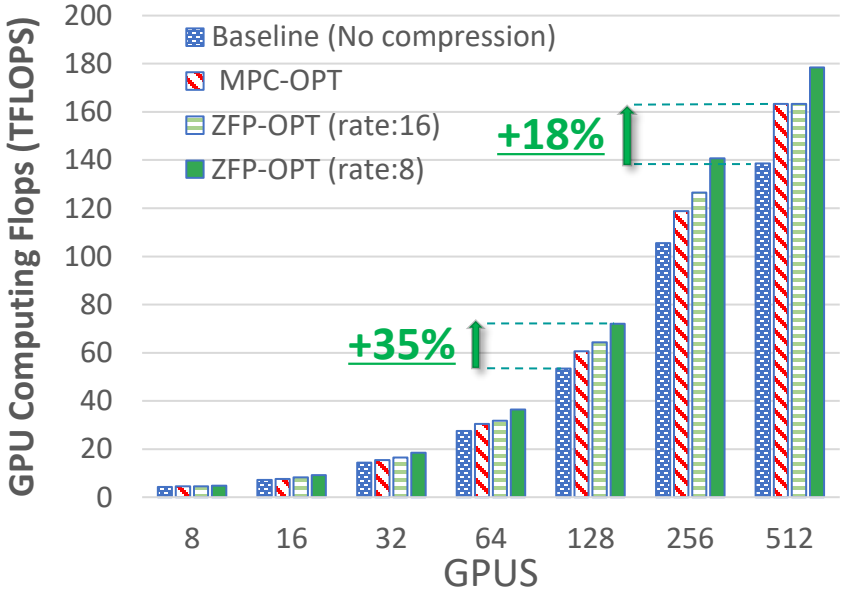

*Platform: ThetaGPU (NVIDIA DGX-A100) (NVIDIA Ampere GPUs connected with NVSwitch), CUDA 11.0*

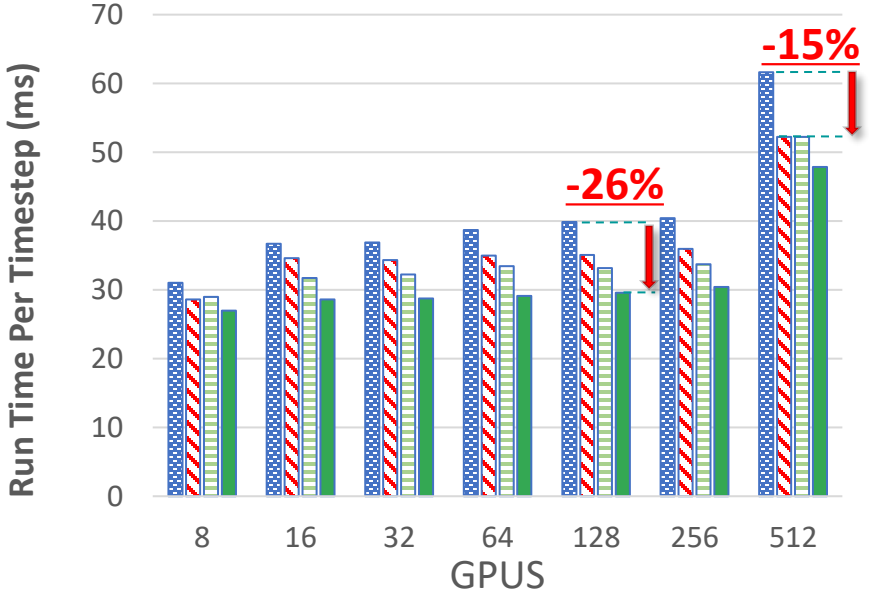
K. Suresh, K. Khorassani, C. Chen, B. Ramesh, M. Abduljabbar, A. Shafi, D. Panda, Network Assisted Non-Contiguous Transfers for GPU-Aware MPI Libraries, Hot Interconnects 29

More details in tomorrow's talk by Kaushik Kandadi Suresh

# “On-the-fly” Compression Support in MVAPICH2-GDR

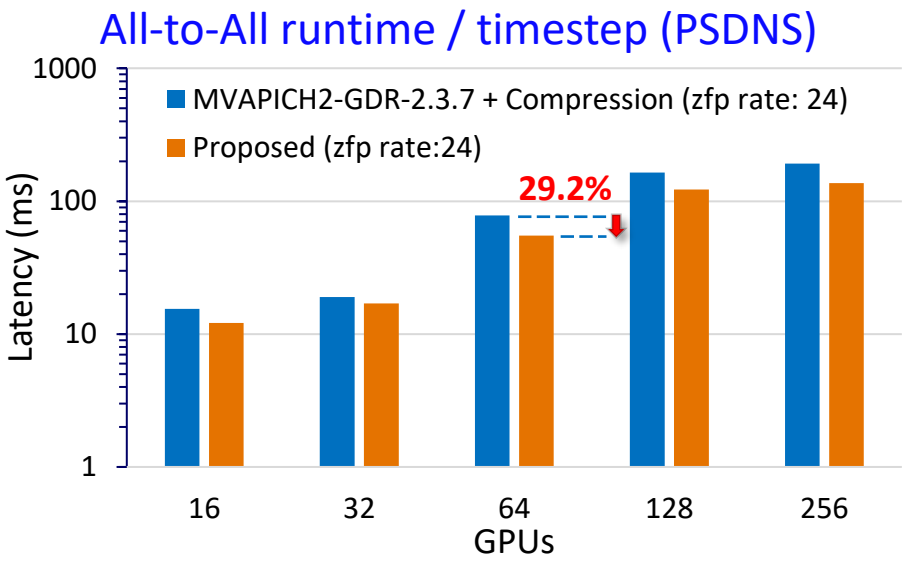
- Weak-Scaling of HPC application **AWP-ODC** on Lassen cluster (V100 nodes)
- MPC-OPT achieves up to **+18%** GPU computing flops, **-15%** runtime per timestep
- ZFP-OPT achieves up to **+35%** GPU computing flops, **-26%** runtime per timestep

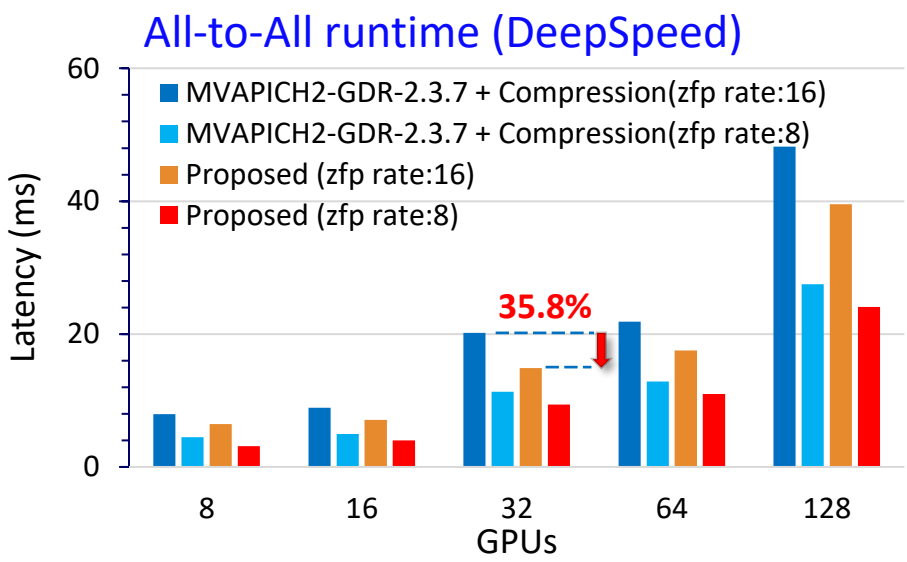



Q. Zhou, C. Chu, N. Senthil Kumar, P. Kousha, M. Ghazimirsaeed, H. Subramoni, and D.K. Panda, Designing High-Performance MPI Libraries with On-the-fly Compression for Modern GPU Clusters, 35th IEEE International Parallel & Distributed Processing Symposium (IPDPS), May 2021. [Best Paper Finalist]

# Performance of All-to-All with Online Compression



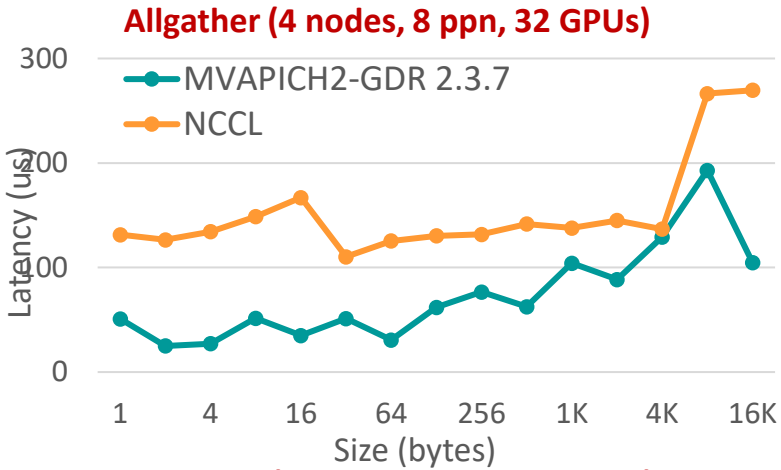



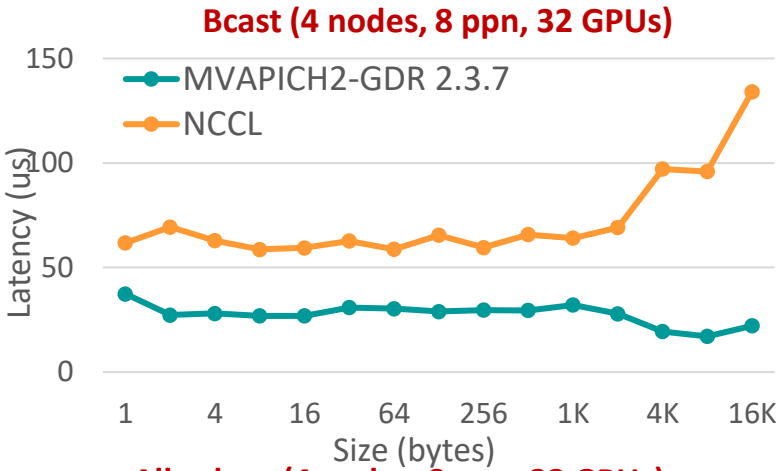

- Improvement compared to MVAPICH2-GDR-2.3.7 with Point-to-Point compression
  - 3D-FFT: Reduce All-to-All runtime by up to 29.2% with ZFP(rate: 24) on 64 GPUs
  - DeepSpeed benchmark: Reduce All-to-All runtime by up to 35.8% with ZFP(rate: 16) on 32 GPUs

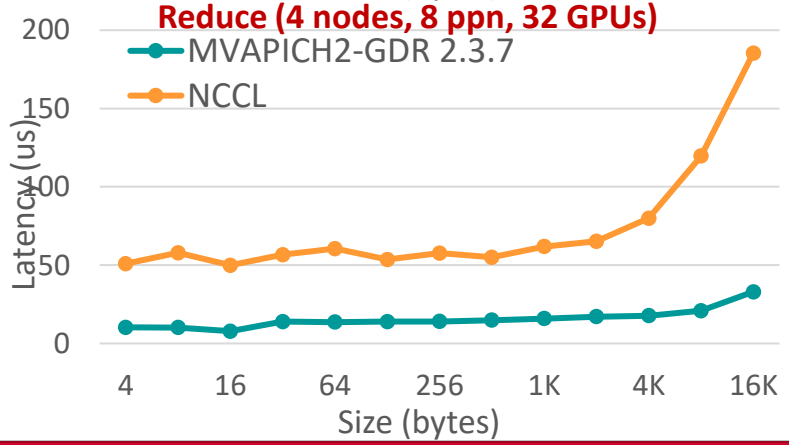
Q. Zhou, P. Kousha, Q. Anthony, K. Khorassani, A. Shafi, H. Subramoni, and D.K. Panda, "Accelerating MPI All-to-All Communication with Online Compression on Modern GPU Clusters", ISC '22.

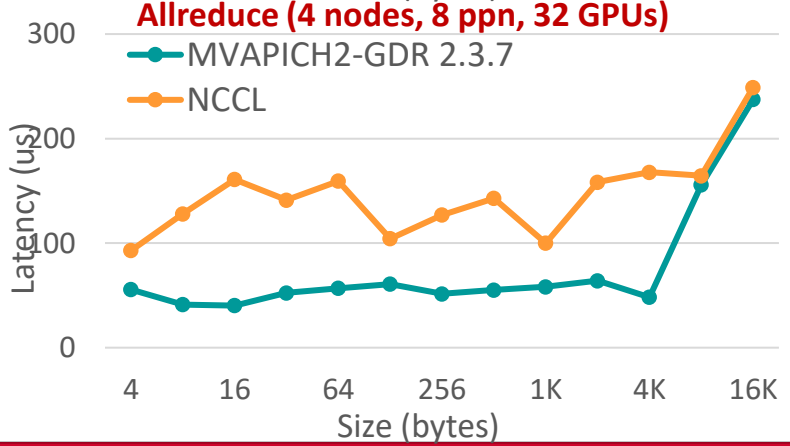
Available in MVAPICH2-GDR 2.3.7

# Collectives Performance on DGX2-A100 – Small Message

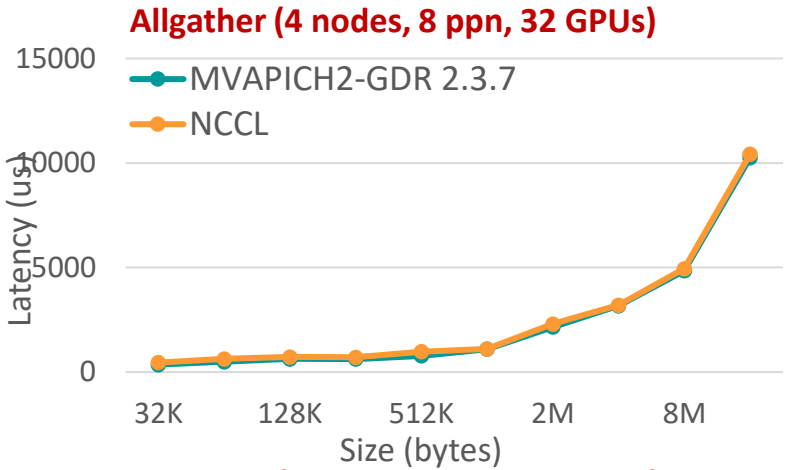


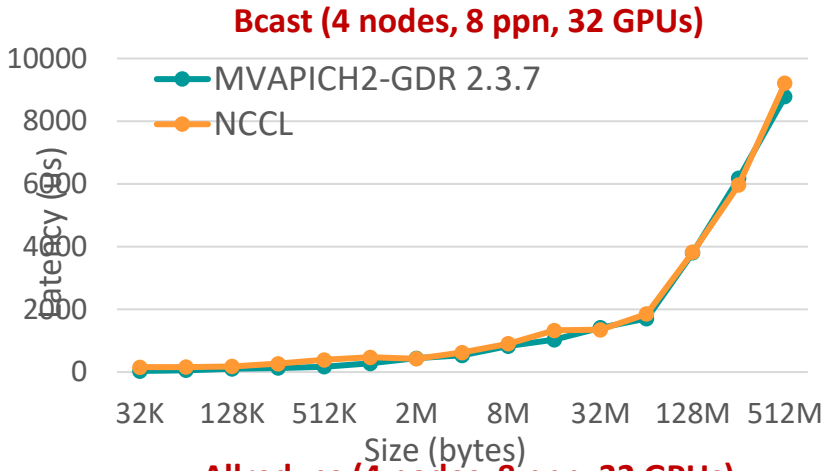



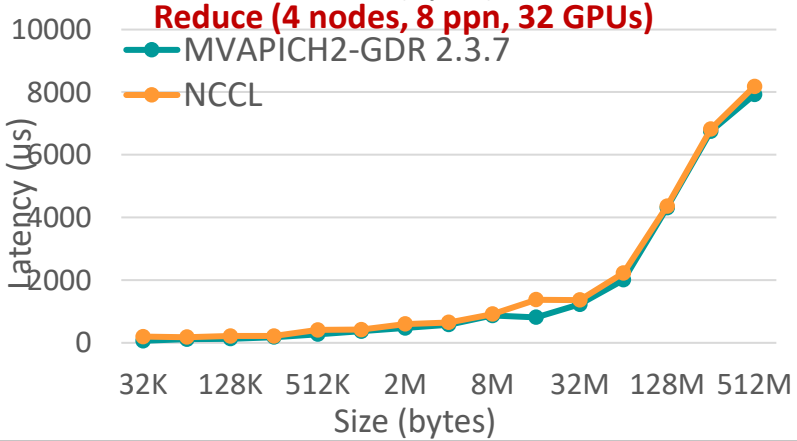



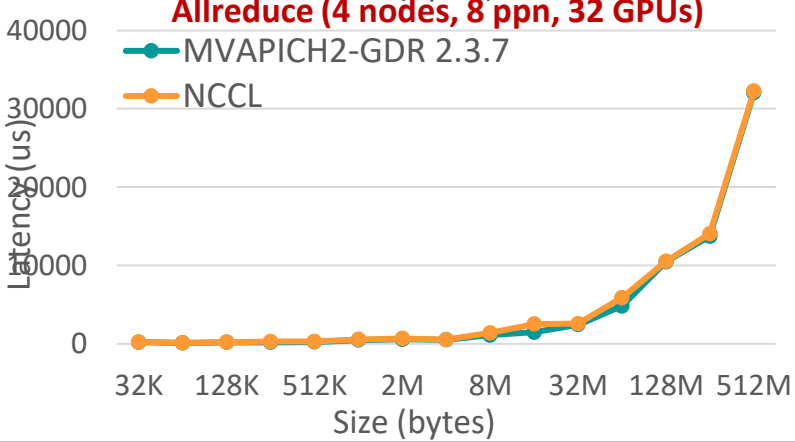



# Collectives Performance on DGX2-A100 – Large Message



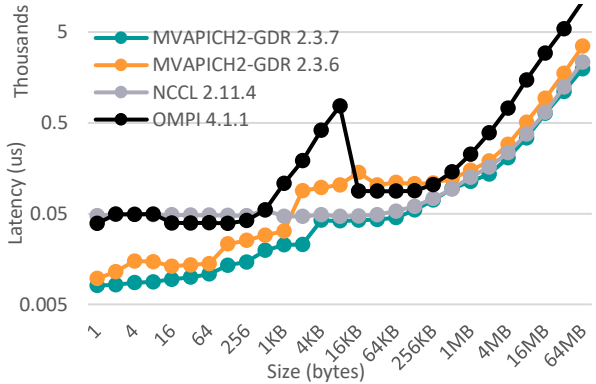



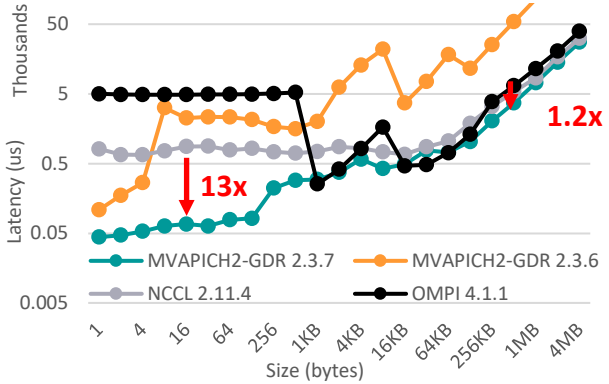
# Highly Efficient Optimized Alltoall(v) Communication

Propose an optimized Alltoall(v) design to overlap inter (sendrecv-based) and intra-node (IPC-based) communication.

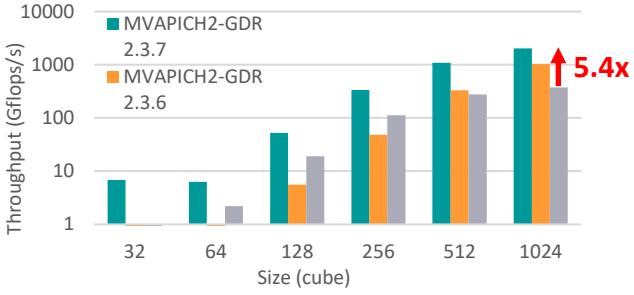
## Alltoall latency on 1 node (8 GPUs)



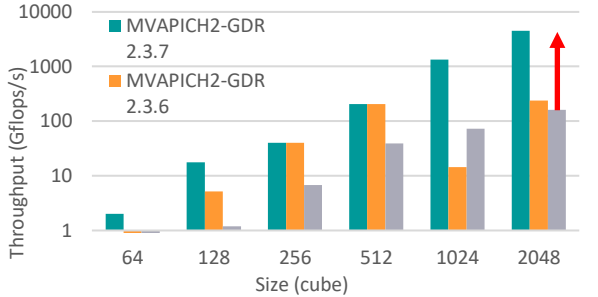

## Alltoall latency on 16 nodes (128 GPUs)




## heFFTe throughput (alltoallv) on 1 node (8 GPUs)




## heFFTe throughput (alltoallv) on 16 node (128 GPUs)

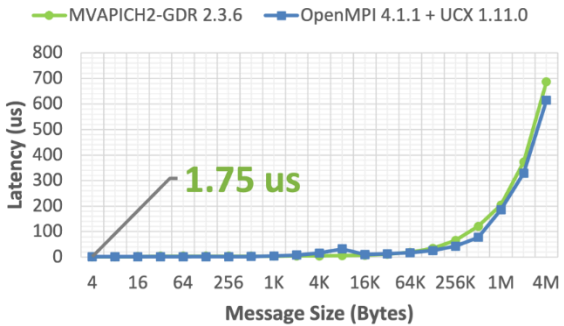



**Available in MVAPICH2-GDR v2.3.7**

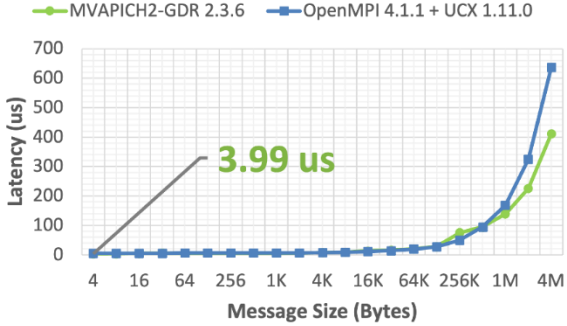
C.-C. Chen, K. Shafie Khorassani, Q. Anthony, A. Shafi, H. Subramoni and D. Panda, "Highly Efficient Alltoall and Alltoallv Communication Algorithms for GPU Systems", HCW '22

# ROCm-aware MVAPICH2-GDR - Support for AMD GPUs

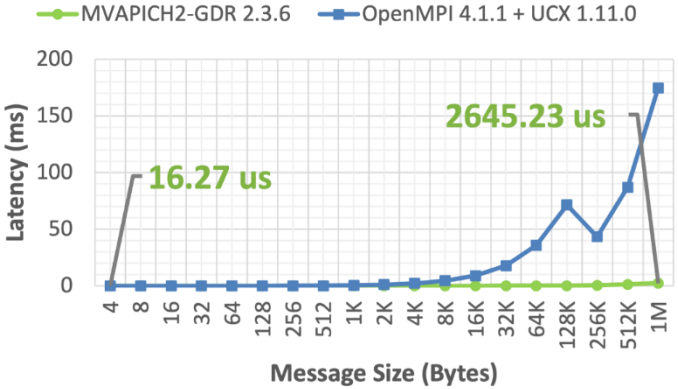
### Intra-Node Point-to-Point Latency



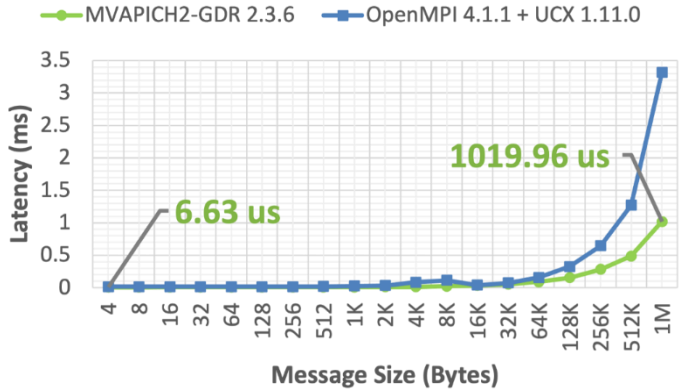

### Inter-Node Point-to-Point Latency




### Allreduce – 64 GPUs (8 nodes, 8 GPUs Per Node)




### Bcast – 64 GPUs (8 nodes, 8 GPUs Per Node)




**Corona Cluster @ LLNL - ROCm-4.3.0 (mi50 AMD GPUs)**

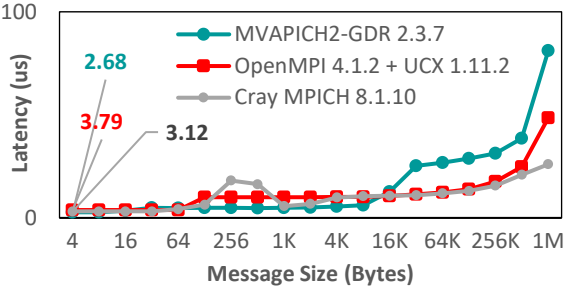
**Available with MVAPICH2-GDR 2.3.5+ & OMB v5.7+**

K. Khorassani, J. Hashmi, C. Chu, C. Chen, H. Subramoni, D. Panda Designing a ROCm-aware MPI Library for AMD GPUs: Early Experiences - ISC HIGH PERFORMANCE 2021, Jun 2021.

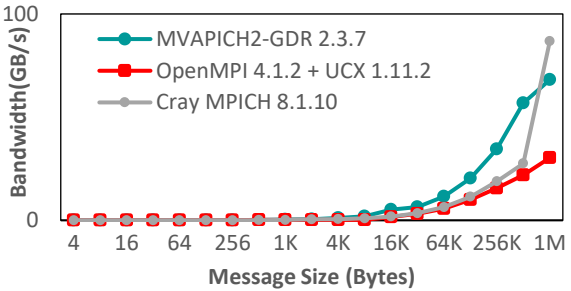
# MVAPICH2-GDR on Slingshot-10 - GPU

## Point-to-Point – Intra-Node

### Latency:

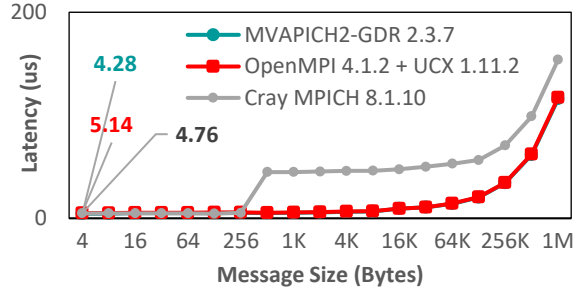



### Bandwidth:

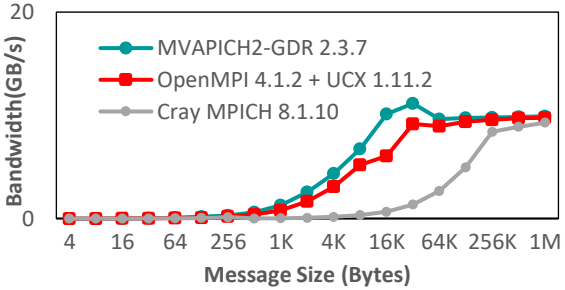



## Point-to-Point – Inter-Node

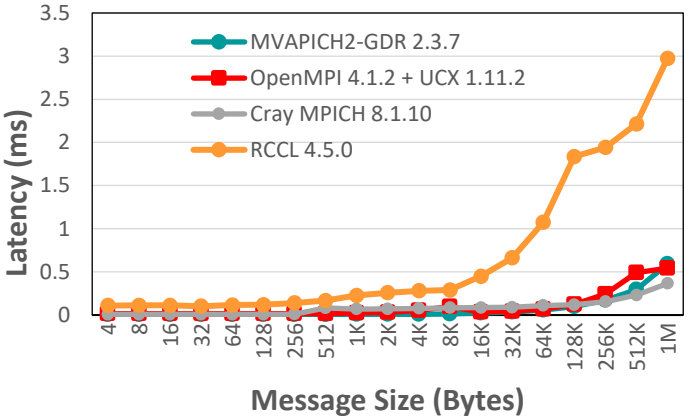
### Latency:



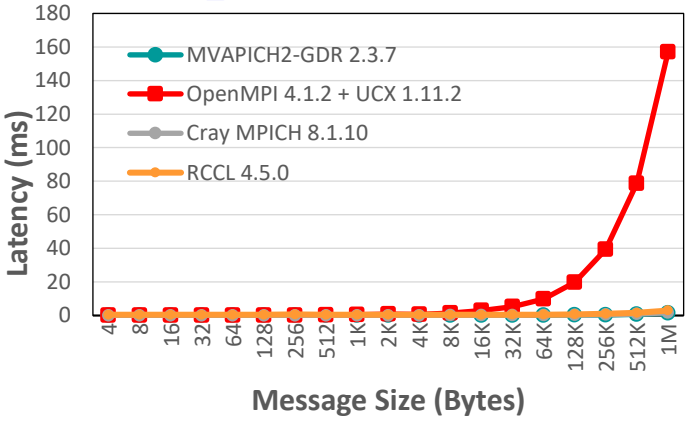

### Bandwidth:




## MPI_Bcast (4 Nodes, 64PPN)

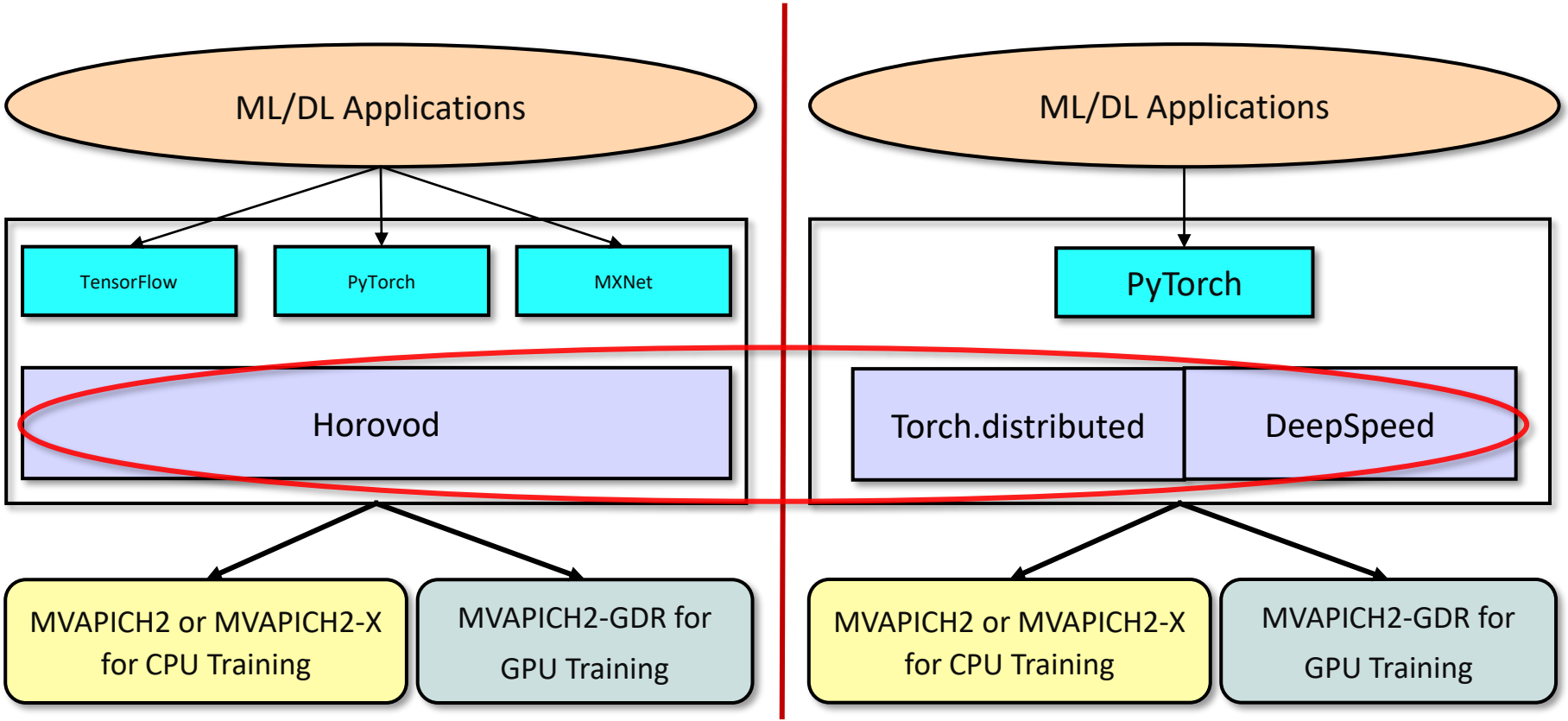



## MPI_Allreduce (4 Nodes, 64PPN)




## AMD Epyc Rome CPUs and AMD MI100 GPUs

# MVAPICH2 (MPI)-driven Infrastructure for ML/DL Training




**More details available from:** http://hidl.cse.ohio-state.edu

# PyTorch, Horovod and DeepSpeed at Scale: Training ResNet-50 on 256 V100 GPUs

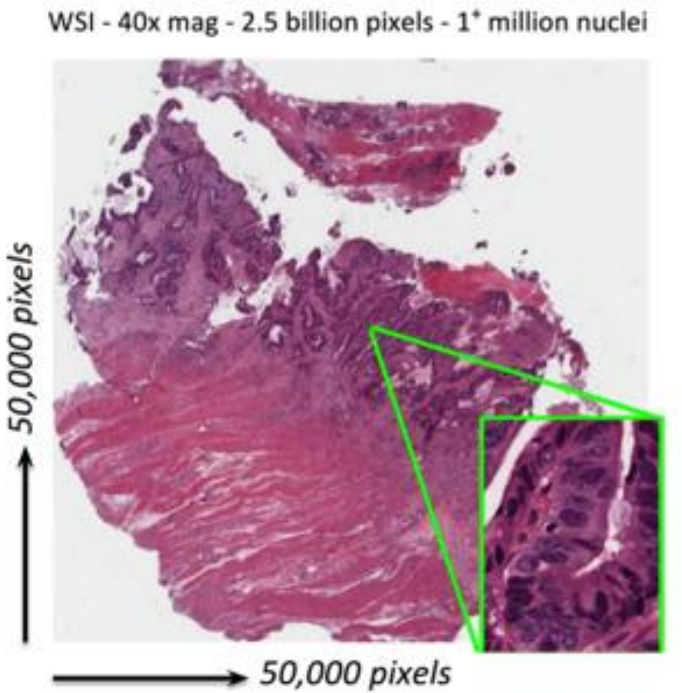
- Training performance for 256 V100 GPUs on LLNL Lassen
  - **~10,000 Images/sec faster** than NCCL training!

| Distributed Framework | Torch.distributed | | Horovod | | DeepSpeed | |
|---|---|---|---|---|---|---|
| Images/sec on 256 GPUs | 61,794 | **72,120** | 74,063 | **84,659** | 80,217 | **88,873** |
| Communication Backend | NCCL | MVAPICH2-GDR | NCCL | MVAPICH2-GDR | NCCL | MVAPICH2-GDR |

# Exploiting Model Parallelism in AI-Driven Digital Pathology

**More details in Tomorrow's talk by Arpan Jain**

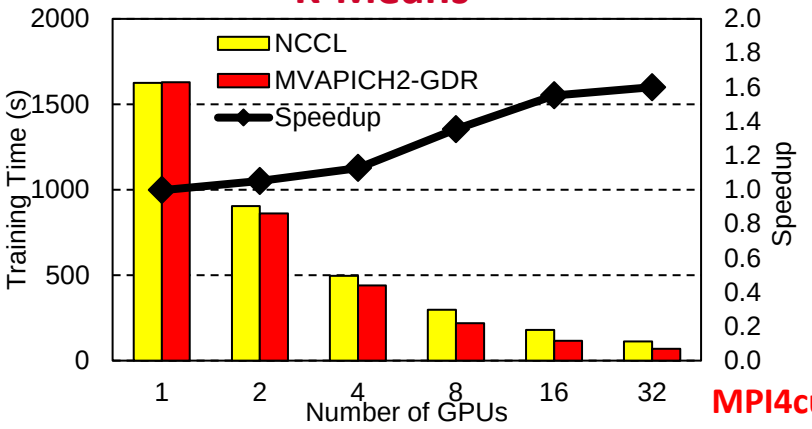
- Pathology whole slide image (WSI)
  - Each WSI = 100,000 x 100,000 pixels
  - Can not fit in a single GPU memory
  - Tiles are extracted to make training possible
- Two main problems with tiles
  - Restricted tile size because of GPU memory limitation
  - Smaller tiles loose structural information
- Can we use Model Parallelism to train on larger tiles to get better accuracy and diagnosis?
- Reduced training time significantly on OpenPOWER + NVIDIA V100 GPUs
  - **32 hours (1 node, 1 GPU) -> 7.25 hours (1 node, 4 GPUs) -> 27 mins (32 nodes, 128 GPUs)**



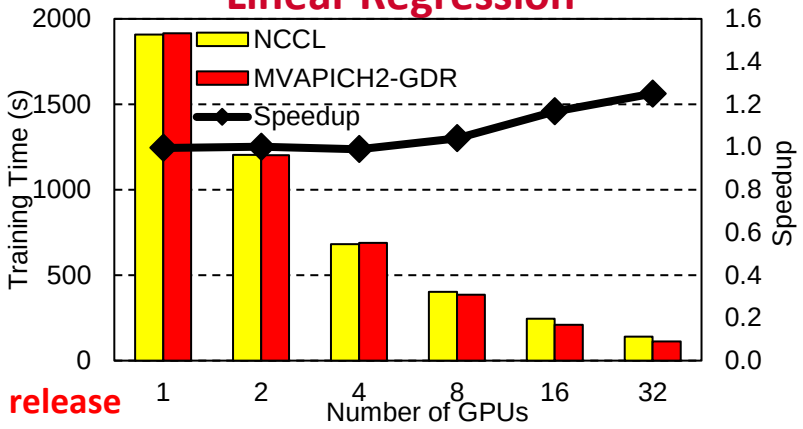

Courtesy: https://blog.kitware.com/digital-slide-archive-large-image-and-histomicstk-open-source-informatics-tools-for-management-visualization-and-analysis-of-digital-histopathology-data/

A. Jain, A. Awan, A. Aljuhani, J. Hashmi, Q. Anthony, H. Subramoni, D. K. Panda, R. Machiraju, and A. Parwani, "GEMS: GPU Enabled Memory Aware Model Parallelism System for Distributed DNN Training", Supercomputing (SC '20)

## K-Means



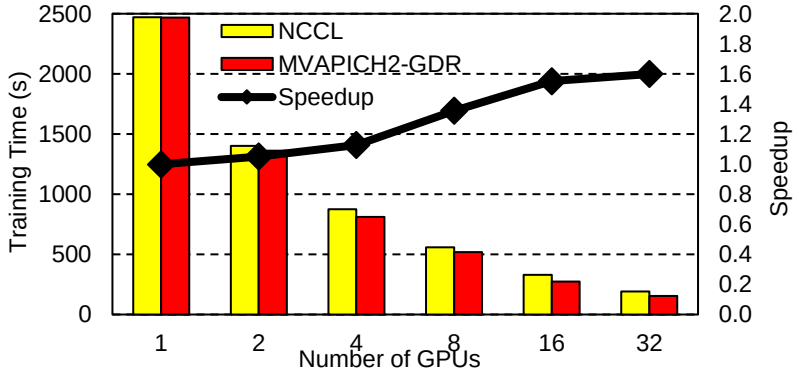

## Linear Regression



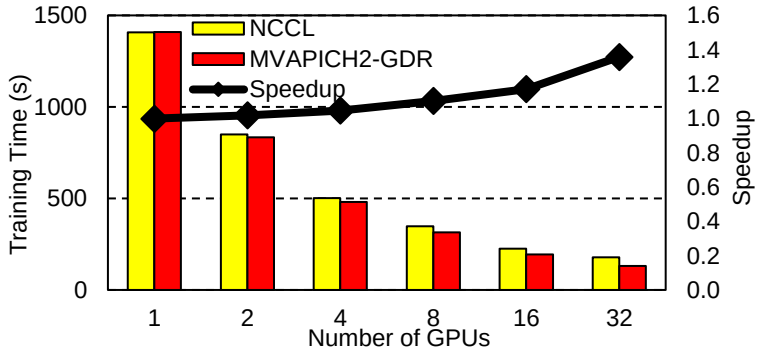

**MPI4cuML 0.1 release**

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

## Nearest Neighbors



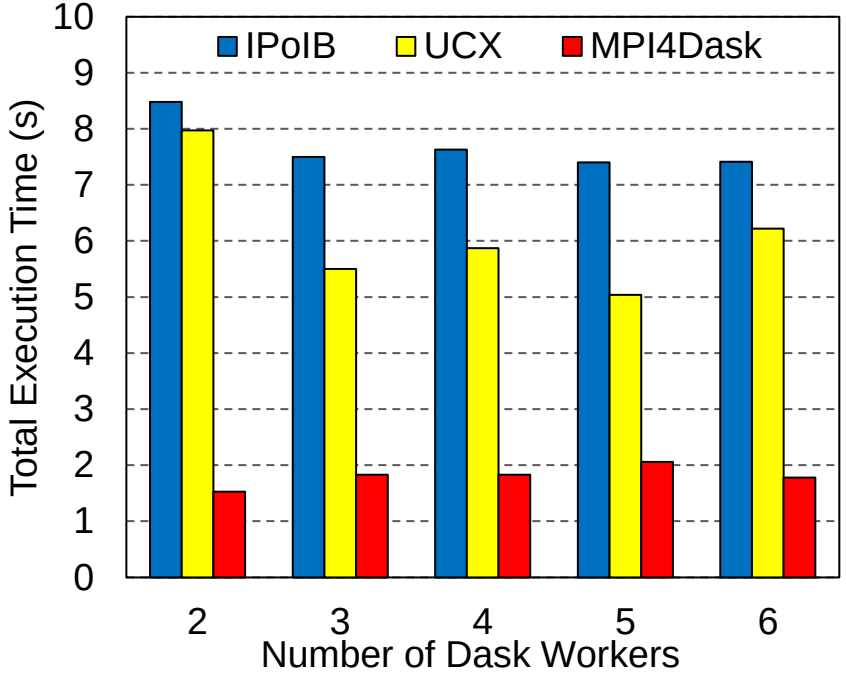

## Truncated SVD



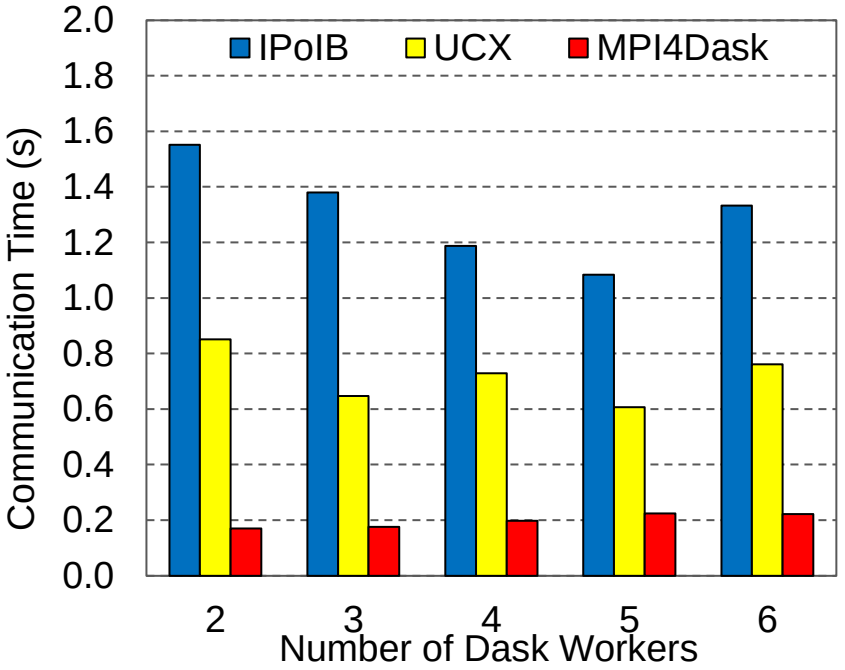

M. Ghazimirsaeed , Q. Anthony , A. Shafi , H. Subramoni , and D. K. Panda, Accelerating GPU-based Machine Learning in Python using MPI Library: A Case Study with MVAPICH2-GDR, MLHPC Workshop, Nov 2020

# Benchmark #1: Sum of cuPy Array and its Transpose (RI2)

**3.47x better on average**




**6.92x better on average**




A. Shafi , J. Hashmi , H. Subramoni , and D. K. Panda, Efficient MPI-based Communication for GPU-Accelerated Dask Applications, CCGrid '21, https://arxiv.org/abs/2101.08878

**MPI4Dask 0.2 release**

**(http://hibd.cse.ohio-state.edu)**

# MVAPICH2-GDR Upcoming Features for HPC and DL

- On-the-fly Compression for All_Gather Collective
- Scalable Distributed Training with Model-/Hybrid Parallelism for out-of-core DNN Models
- Scaling Single-Image Super-Resolution Training

# MVAPICH2 Software Family

| Requirements | Library |
|---|---|
| MPI with Support for InfiniBand, Omni-Path, Ethernet/iWARP and, RoCE (v1/v2) | MVAPICH2 |
| Optimized Support for Microsoft Azure Platform with InfiniBand | MVAPICH2-Azure |
| Advanced MPI features/support (UMR, ODP, DC, Core-Direct, SHArP, XPMEM), OSU INAM (InfiniBand Network Monitoring and Analysis) | MVAPICH2-X |
| Advanced MPI features (SRD and XPMEM) with support for Amazon Elastic Fabric Adapter (EFA) | MVAPICH2-X-AWS |
| Optimized MPI for clusters with NVIDIA GPUs and for GPU-enabled Deep Learning Applications | MVAPICH2-GDR |
| Energy-aware MPI with Support for InfiniBand, Omni-Path, Ethernet/iWARP and, RoCE (v1/v2) | MVAPICH2-EA |
| MPI Energy Monitoring Tool | OEMT |
| InfiniBand Network Analysis and Monitoring | OSU INAM |
| Microbenchmarks for Measuring MPI and PGAS Performance | OMB |

# Outline

- Brief Overview of the MVAPICH2 Project
- **Features of Recent Releases**
  - MVAPICH2 2.3.7
  - MVAPICH2 3.0a
  - MVAPICH2-J
  - MVAPICH2-X-AWS 2.3.7 and Cloud Deployments
  - MVAPICH2-GDR 2.3.7 and Support for ML, DL, and Data Science
  - **OMB 6.0**
  - Applications: Best Practices

# OSU Microbenchmarks

- Available since 2004
- Suite of microbenchmarks to study communication performance of various programming models and programming languages
- Benchmarks available for the following programming models
  - Message Passing Interface (MPI)
  - Partitioned Global Address Space (PGAS)
    - Unified Parallel C (UPC), Unified Parallel C++ (UPC++), and OpenSHMEM
- Benchmarks available for multiple programming languages
  - C, Java, Python
- Benchmarks available for multiple accelerator-based architectures
  - Compute Unified Device Architecture (CUDA)
  - OpenACC Application Program Interface
  - Radeon Open Compute software platform (ROCm)
- Part of various national resource procurement suites like NERSC-8 / Trinity Benchmarks
- Continuing to add support for newer primitives and features (latest OMB 7.0)
  - Integrated support for plotting graphs, histograms and profiling using PAPI.
- Please visit the following link for more information: http://mvapich.cse.ohio-state.edu/benchmarks/

# Java and Python Extensions to OMB

- Java and Python extensions have been released as part of the OMB 6.0 release:
  - https://mvapich.cse.ohio-state.edu/benchmarks/
- Instructions for using OMB for Java:
  - User guide: https://mvapich.cse.ohio-state.edu/static/media/mvapich/README-OMB-J.txt
  - Sample run:

```
mpirun_rsh -np 2 -hostfile hosts \
  LD_PRELOAD=${MPILIB}/lib/libmpi.so java -cp $MV2J_HOME/lib/mvapich2-j.jar:. \
  -Djava.library.path=$MV2J_HOME/lib mpi.pt2pt.OSUBandwidth
```

- Instructions for using OMB for Python:
  - User guide: https://mvapich.cse.ohio-state.edu/static/media/mvapich/README-OMB-PY.txt
  - Sample run:

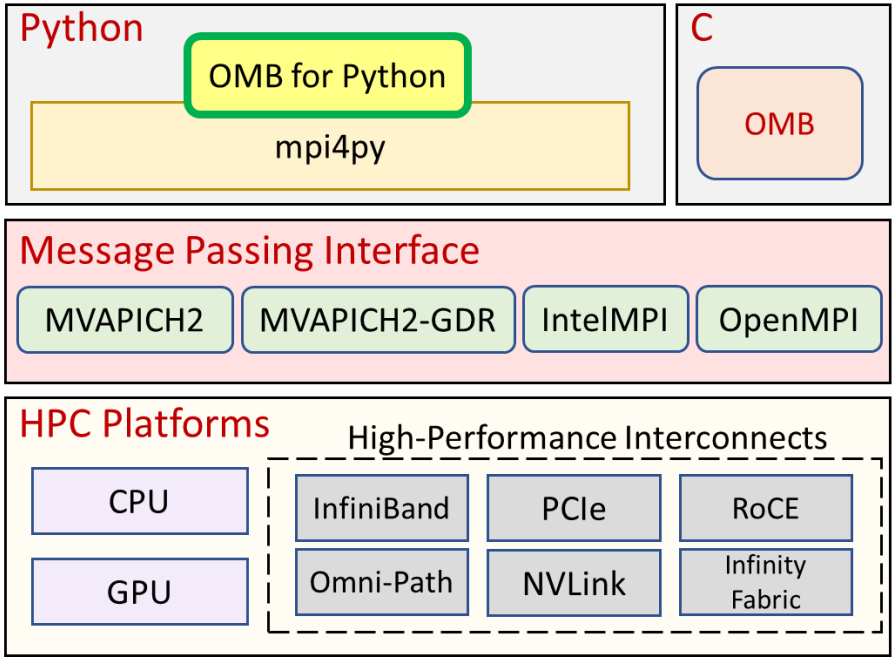
```
mpirun -np 2 --hostfile hosts python run.py \
--benchmark latency --buffer numpy
```

# Java extensions to OMB

- Java extensions to OMB include:
  - point-to-point primitives (latency, bandwidth, and bi-bandwidth)
  - vectored and blocking collective communication primitives (latency)

| Point-to-Point | Collectives | Vectored Collectives |
|---|---|---|
| OSULatency | OSUAllgather | OSUAllgatherv |
| OSUBandwidth | OSUAlltoall | OSUAlltoallv |
| OSUBiBandwidth | OSUGather | OSUGatherv |
| - OSUBandwidthOMPI | OSUScatter | OSUScatterv |
| - OSUBiBandwidthOMPI | OSUReduce | |
| | OSUAllReduce | |
| | OSUBcast | |
| | OSUBarrier | |
| | OSUReduceScatter | |

# Python Benchmarks




Architectural hierarchy of OMB for Python with mpi4py, MPI, and HPC platforms

| | |
|---|---|
| Point-to-Point | Bi-directional bandwidth, Bandwidth, Latency, Multi latency |
| Blocking Collectives | Allgather, Allreduce, Alltoall, Barrier, Bcast, Gather, Reduce_scatter, scatter |
| Vector Variant | Allgatherv, Alltoallv, Gatherv, Scatterv |

Point-to-Point, blocking collectives, and vector variant benchmarks supported by OMB for Python

N. Alnaasan , A. Jain , A. Shafi , H. Subramoni , and DK Panda, OMB-Py: Python Micro-Benchmarks for Evaluating Performance of MPI Libraries on HPC Systems

# Package Comparison

- Python extensions to the open-source OMB suite aimed to evaluate communication performance of MPI-based parallel applications in Python.
- Performance characterization of MPI communication in Python on four HPC systems:
  - Point-to-point and collective communication operations using OMB as a baseline performance in C.
  - Evaluation on CPU and GPU devices for different buffers including Bytearrays, Numpy, CuPy, PyCUDA and Numba.
  - Pickle method evaluation for serializing communicated objects.
- Analysis of the overhead presented by mpi4py over native MPI libraries.

| | Point-to-point | Blocking Collectives | Vector Variants | Support for Python | Bytearray Buffers | Numpy Buffers | CuPy Buffers | PyCUDA Buffers |
|---|---|---|---|---|---|---|---|---|
| OMB for Python | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| mpi4py Demo Codes [1] | ✓ | Partial | Partial | ✓ | ✗ | ✓ | ✗ | ✗ |
| IMB [2] | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ |
| SMB [3] | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |

**Feature Comparison Between Benchmark Packages**

[1] L. Dalcin, R. Paz, and M. Storti, MPI for Python, Journal of Parallel and Distributed Computing, 65(9):1108-1115, 2005. https://doi.org/10.1016/j.jpdc.2005.03.010
[2] "Sandia MPI Micro-Benchmark Suite (SMB)." http://www.cs.sandia.gov/smb/index.html
[3] "Intel MPI Benchmarks (IMB)." https://software.intel.com/en-us/articles/intel-mpi-benchmarks

N. Alnaasan , A. Jain , A. Shafi , H. Subramoni , and DK Panda, OMB-Py: Python Micro-Benchmarks for Evaluating Performance of MPI Libraries on HPC Systems

# Outline

- Brief Overview of the MVAPICH2 Project
- **Features of Recent Releases**
  - MVAPICH2 2.3.7
  - MVAPICH2 3.0a
  - MVAPICH2-J
  - MVAPICH2-X-AWS 2.3.7 and Cloud Deployments
  - MVAPICH2-GDR 2.3.7 and Support for ML, DL, and Data Science
  - OMB 6.0
  - **Applications: Best Practices**

# 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.

# MVAPICH2 – Future Roadmap and Plans for Exascale

- Making CH4 channel default
  - Early 2023
- Performance and Memory scalability toward 1M-10M cores
- Hybrid programming (MPI + OpenSHMEM, MPI + UPC, MPI + CAF …)
  - MPI + Task*
- Enhanced Optimization for GPUs and FPGAs*
- Taking advantage of advanced features of Mellanox InfiniBand
  - Tag Matching*
  - Adapter Memory*
- Enhanced communication schemes for upcoming architectures
  - NVLINK*
  - CAPI*
  - Bluefield2*
- Extended topology-aware collectives
- Extended Energy-aware designs and Virtualization Support
- Extended Support for MPI Tools Interface (as in MPI 3.0)
- Extended FT support
- Support for * features will be available in future MVAPICH2 Releases

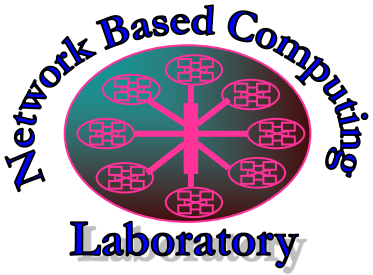
# Join us for Multiple Events at SC '22

- Presentations at OSU and X-Scale Booth (#4305)
  - Members of the MVAPICH, HiBD and HiDL members
  - External speakers
- Presentations at SC main program (Tutorials, Workshops, BoFs, Posters, and Doctoral Showcase)
- Presentation at many other booths (Mellanox, Intel, Microsoft, and AWS) and satellite events
- Complete details available at

  **http://mvapich.cse.ohio-state.edu/conference/904/talks/**

# 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/