MVAPICH
MPI, PGAS and Hybrid MPI+PGAS Library

Overview of the MVAPICH Project: Latest Status and Future Roadmap

MVAPICH2 User Group (MUG) Conference

by

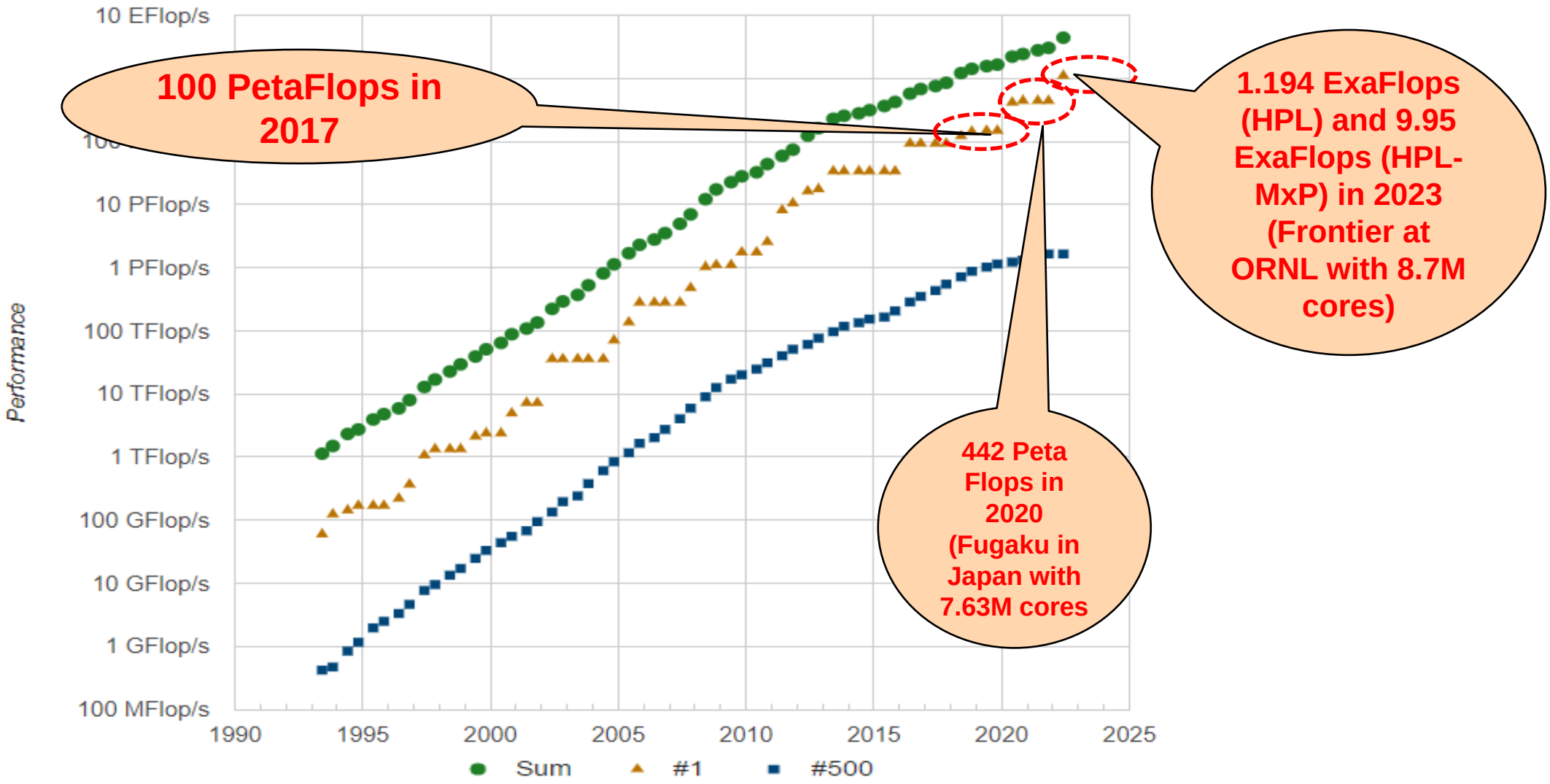
Dhabaleswar K. (DK) Panda

The Ohio State University

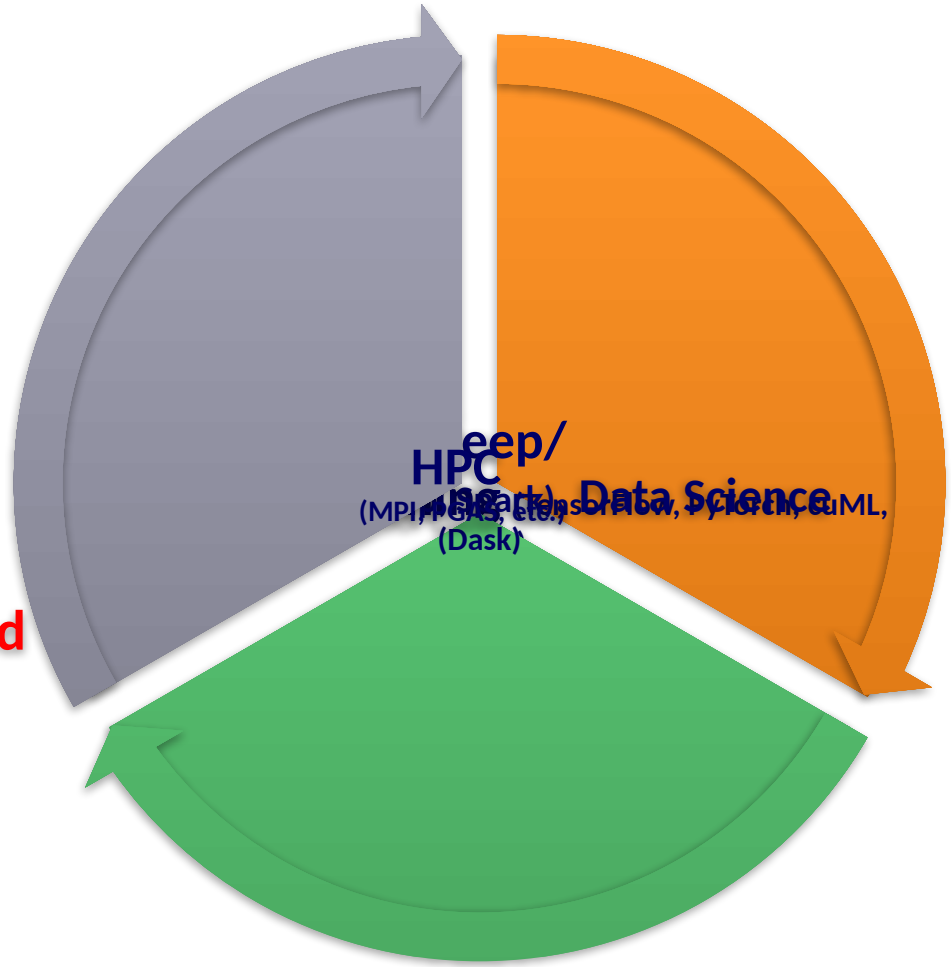
E-mail: panda@cse.ohio-state.edu

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

High-End Computing (HEC): PetaFlop to ExaFlop



Increasing Usage of HPC, AI, Big Data and Data Science



Convergence of HPC, Deep/Machine Learning, and Data Science!

Increasing Need to Run these applications on the Cloud!!

Can MPI-Driven Middleware be designed and used for all three domains?

Outline

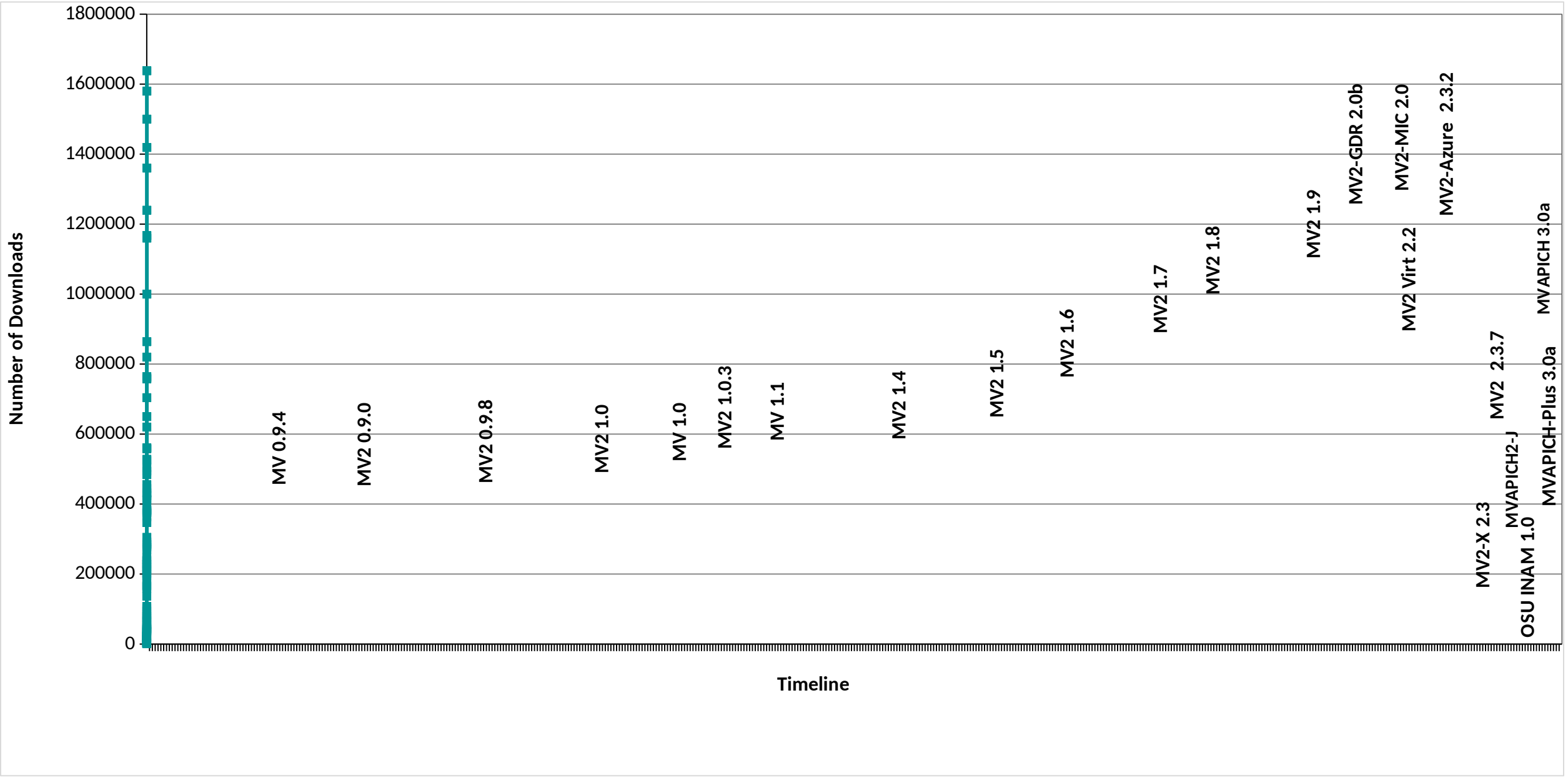
- Brief Overview of the MVAPICH Project
- Features of Recent Releases
  - MVAPICH2 2.3.7
  - MVAPICH 3.0b
  - MVAPICH2-J
  - MVAPICH2-X, MVAPICH2-X-AWS 2.3.7 and Cloud Deployments
  - MVAPICH2-GDR 2.3.7 for GPUs
  - MVAPICH-Plus 3.0a2
  - Converged software stack based on MVAPICH
    - Support for DL (HiDL), ML (MPI4cuML), Big Data (MPI4Spark), and Data Science (MPI4Dask)
  - OSU Micro-Benchmarks (OMB)
  - InfiniBand Network Analysis and Monitoring (INAM)
  - Applications: Best Practices
- Upcoming Features
  - MVAPICH and MVAPICH-Plus 3.0 series
  - MVAPICH-Plus for AI-Accelerator Chips (Habana/Gaudi)
  - MVAPICH and OMB for FPGA
  - INAM
  - MPI4DL for Deep Learning
  - MPI4Spark
  - Conversational AI Interface (SAI)
- Conclusions

Overview of the MVAPICH Project

- High Performance open-source MPI Library
- Support for multiple interconnects
  - InfiniBand, Omni-Path, Ethernet/iWARP, RDMA over Converged Ethernet (RoCE), AWS EFA, OPX, Broadcom RoCE, Intel Ethernet, Rockport Networks, Slingshot 10/11
- 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 3,325 organizations in 90 countries**
- **More than 1.70 Million downloads from the OSU site directly**
- Empowering many TOP500 clusters (Jun '23 ranking)
  - 7th , 10,649,600-core (Sunway TaihuLight) at NSC, Wuxi, China
  - 21st , 448, 448 cores (Frontera) at TACC
  - 36th, 288,288 cores (Lassen) at LLNL
  - 49th, 570,020 cores (Nurion) in South Korea and many others
- Available with software stacks of many vendors and Linux Distros (RedHat, SuSE, OpenHPC, and Spack)
- Partner in the 21st ranked TACC Frontera system
- **Empowering Top500 systems for more than 16 years**

MVAPICH2 Release Timeline and Downloads



Architecture of MVAPICH2 Software Family for HPC, DL/ML, Big Data & Data Science

High Performance Parallel Programming Models

- Message Passing Interface (MPI)
- PGAS (UPC, OpenSHMEM, CAF, UPC++)
- Hybrid --- MPI + X (MPI + PGAS + OpenMP/Cilk)

High Performance and Scalable Communication Runtime

Diverse APIs and Mechanisms

- Point-to-point Primitives
- Collectives Algorithms
- Job Startup
- Energy-Awareness
- Remote Memory Access
- I/O and File Systems
- Fault Tolerance
- Virtualization
- Active Messages
- Introspection & Analysis

Support for Modern Networking Technology (InfiniBand, iWARP, RoCE, Omni-Path, Elastic Fabric Adapter)

Transport Protocols

- RC
- SRD
- UD
- DC

Modern Features

- UMR
- ODP
- SR-IOV
- Multi Rail

Support for Modern Multi-/Many-core Architectures (Intel-Xeon, OpenPOWER, Xeon-Phi, ARM, NVIDIA/AMD GPGPU)

Transport Mechanisms

- Shared Memory
- CMA
- IVSHMEM
- XPMEM

Modern Features

- Optane*
- NVLink
- CAPI*

* 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

Outline

- Brief Overview of the MVAPICH Project
- **Features of Recent Releases**
  - **MVAPICH2 2.3.7**
  - MVAPICH 3.0b
  - MVAPICH2-J
  - MVAPICH2-X-AWS 2.3.7 and Cloud Deployments
  - MVAPICH2-GDR 2.3.7 for GPUs
  - MVAPICH-Plus 3.0a2
  - Converged software stack based on MVAPICH
    - Support for DL (HiDL), ML (MPI4cuML), Big Data (MPI4Spark), and Data Science (MPI4Dask)
  - OSU Micro-Benchmarks (OMB)
  - InfiniBand Network Analysis and Monitoring (INAM)
  - Applications: Best Practices
- Upcoming Features
- Conclusions

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

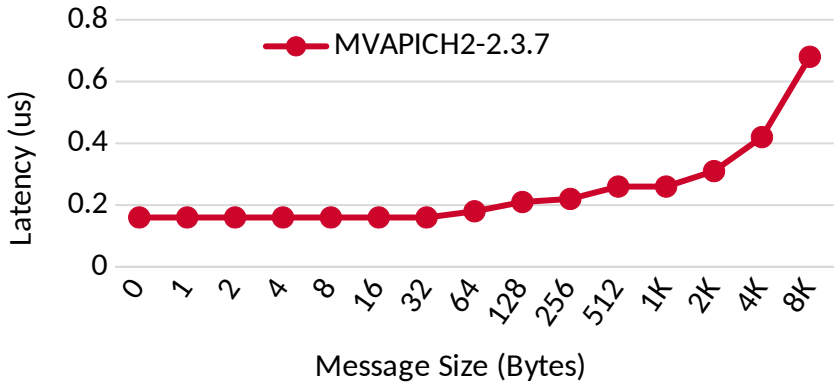
Highlights of MVAPICH2 2.3.7-GA Release

- 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

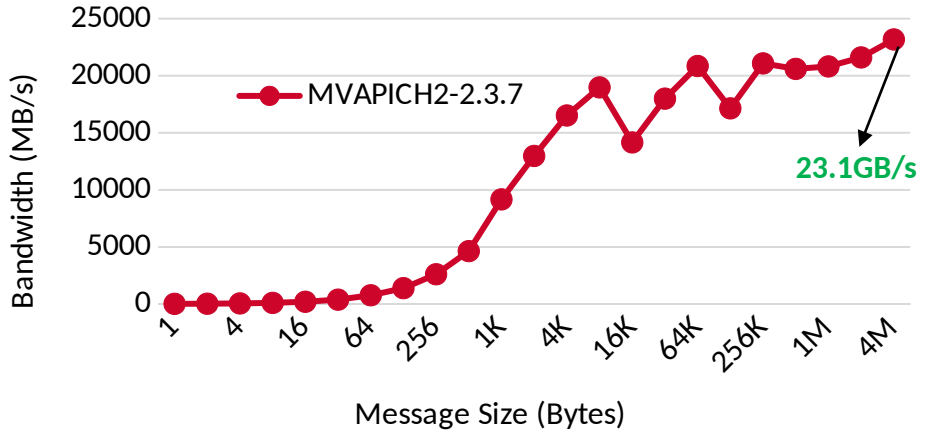
AMD Milan + HDR 200

Intra-Node CPU Point-to-Point

Latency

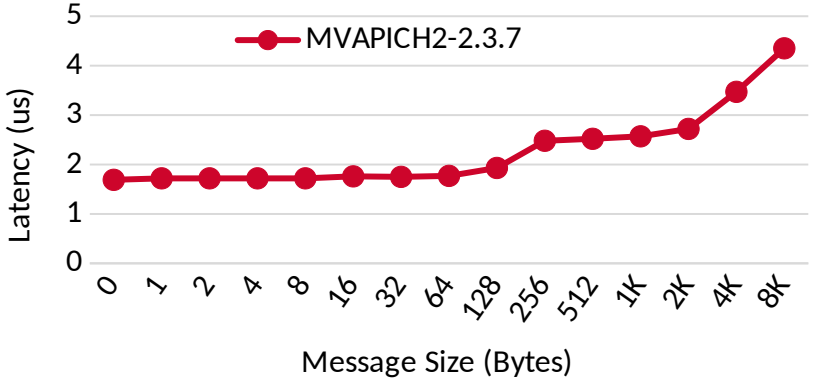


Bandwidth

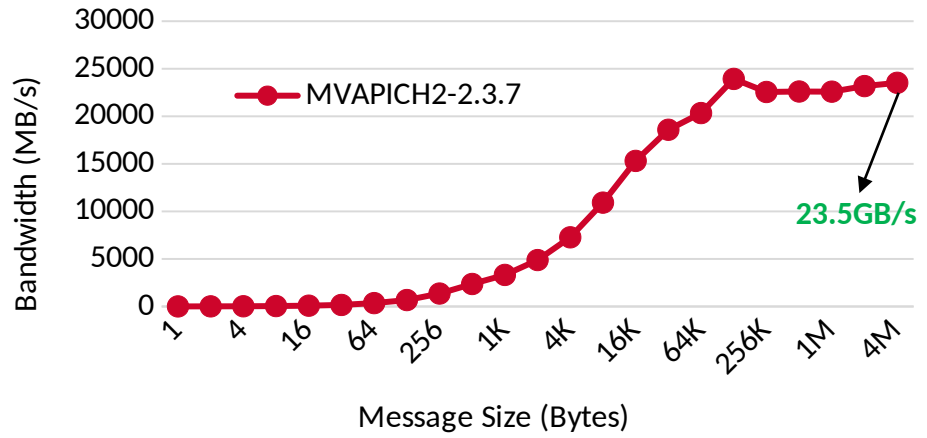


Inter-Node CPU Point-to-Point

Latency

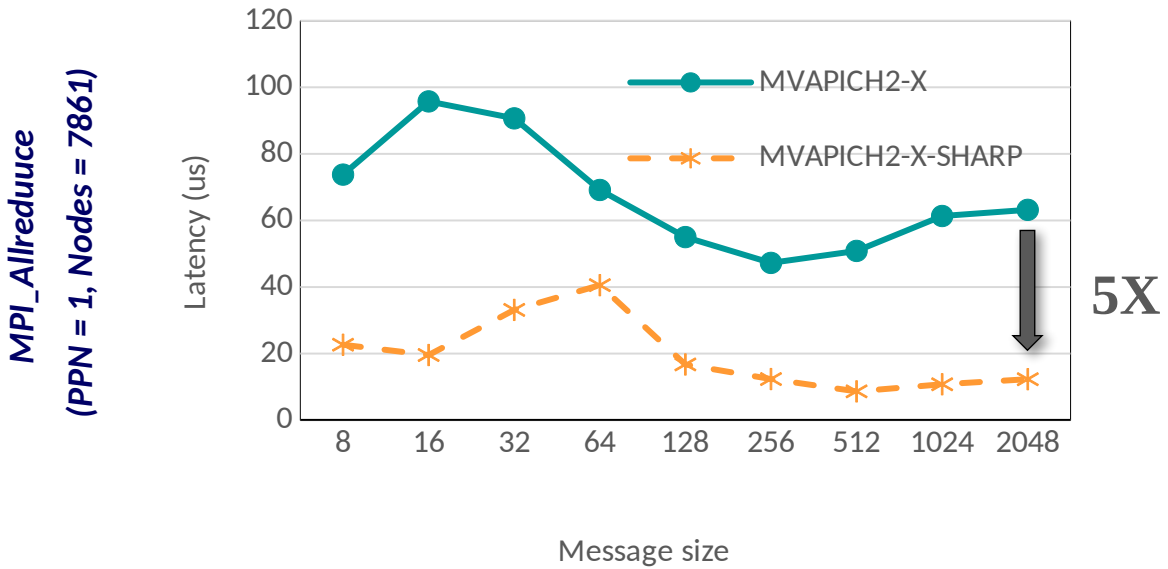


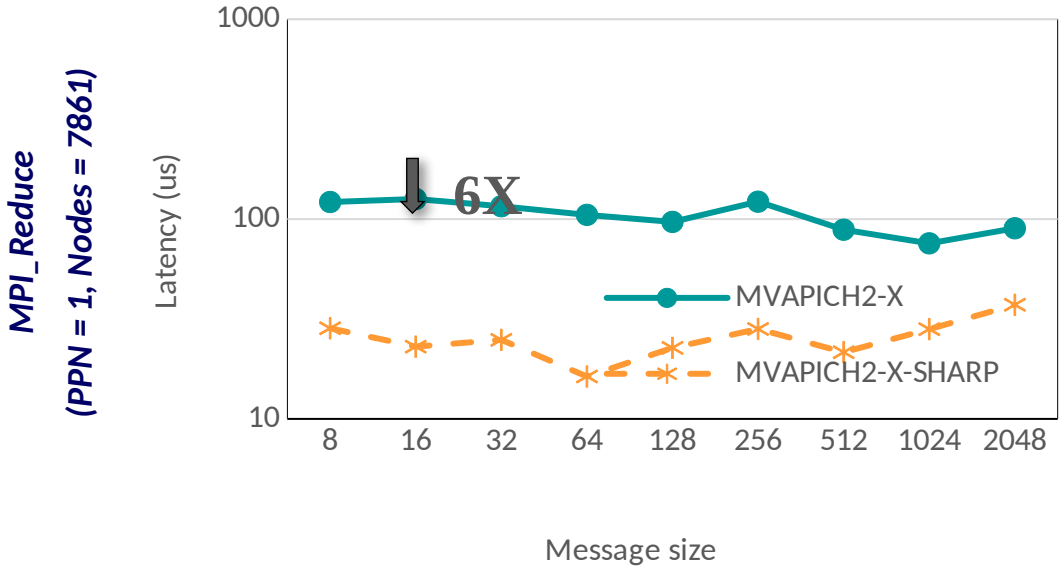
Bandwidth

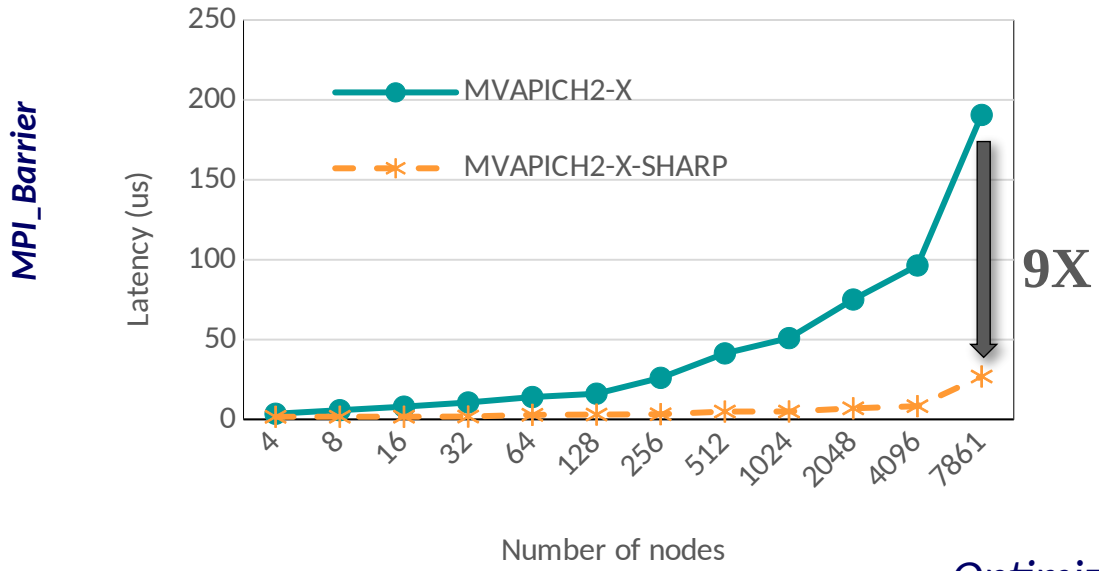


AMD EPYC 7V73X 64-Core Processor, Mellanox ConnectX-6 HDR HCA

Performance of Collectives with SHARP on TACC Frontera







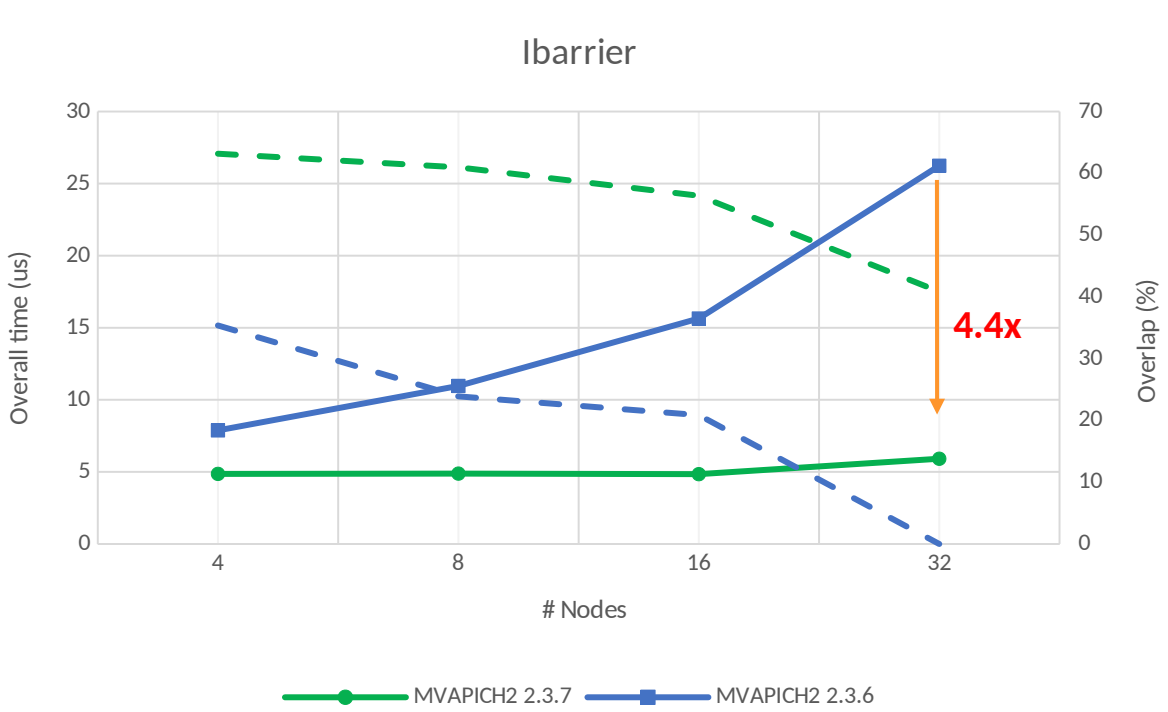
Optimized SHARP designs in MVAPICH2-X

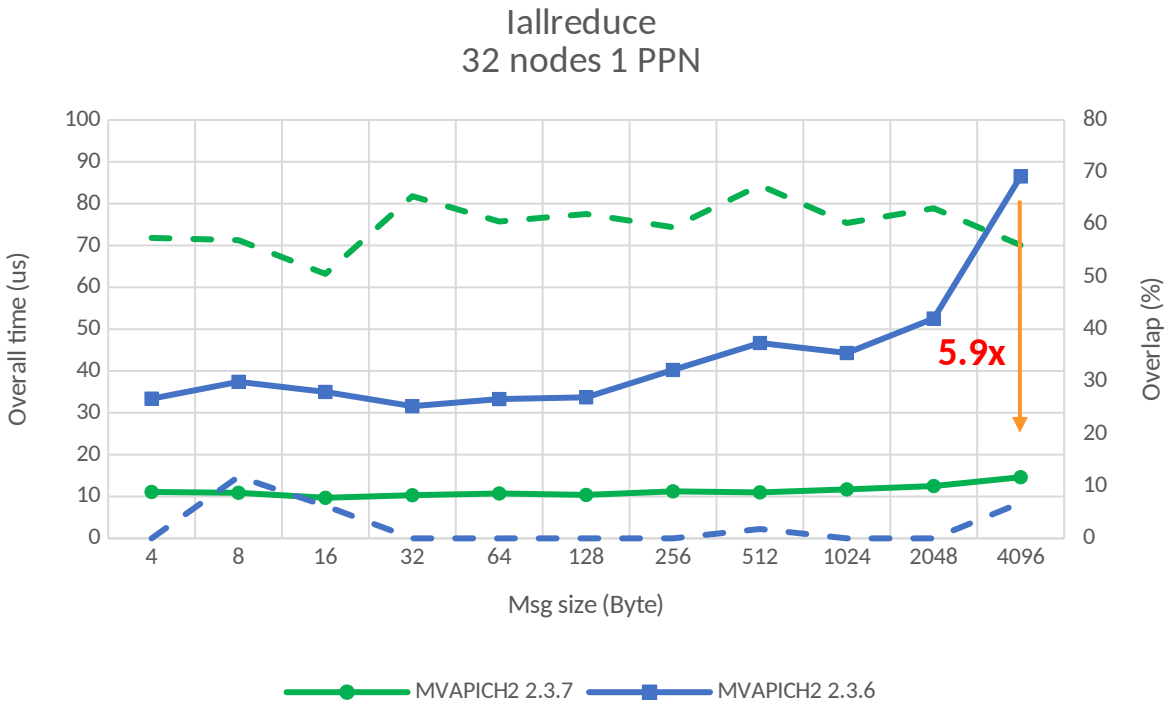
Up to 9X performance improvement with SHARP over MVAPICH2-X default for 1ppn MPI_Barrier, ***6X*** for 1ppn MPI_Reduce and ***5X*** for 1ppn MPI_Allreduce

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



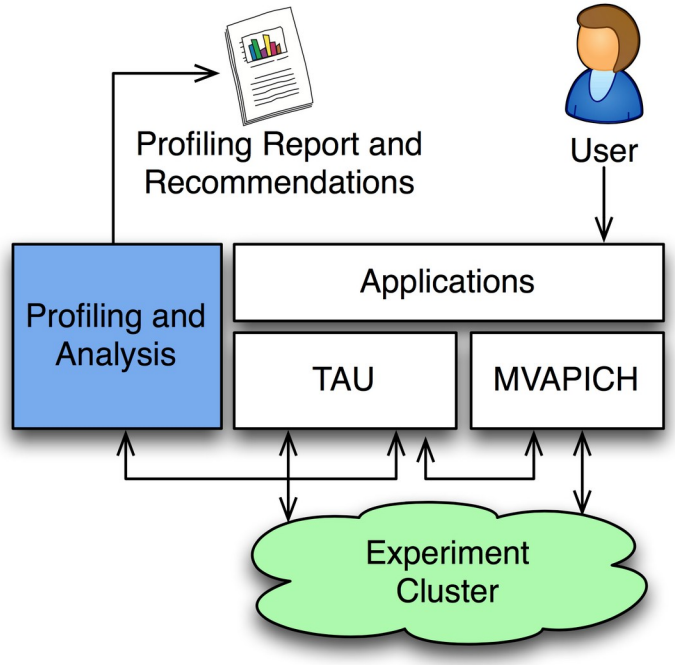


- With SHARP:
  - Flat scaling in terms of overall time
  - High overlap between computation and communication

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

Performance Engineering Applications using MVAPICH2 and TAU

- Enhance existing support for MPI_T in MVAPICH2 to expose a richer set of performance and control variables
- Get and display MPI Performance Variables (PVARs) made available by the runtime in TAU
- Control the runtime's behavior via MPI Control Variables (CVARs)
- Introduced support for new MPI_T based CVARs to MVAPICH2
  - MPIR_CVAR_MAX_INLINE_MSG_SZ, MPIR_CVAR_VBUF_POOL_SIZE, MPIR_CVAR_VBUF_SECONDARY_POOL_SIZE
- TAU enhanced with support for setting MPI_T CVARs in a non-interactive mode for uninstrumented applications
- S. Ramesh, A. Maheo, S. Shende, A. Malony, H. Subramoni, and D. K. Panda, *MPI Performance Engineering with the MPI Tool Interface: the Integration of MVAPICH and TAU*, *EuroMPI/USA '17, Best Paper Finalist*
- More details in Sameer Shende's talk (later today)



VBUF usage without CVAR based tuning as displayed by ParaProf

TAU: ParaProf: Context Events for: node 0 - mpit_withoutcvar_bt.C.1k.ppk

Name	MaxValue	MinValue	MeanValue	Std. Dev.	NumSamples	Total
mv2_total_vbuf_memory (Total amount of memory in bytes used for VBUFs)	3,313,056	3,313,056	3,313,056	0	1	3,313,056
mv2_ud_vbuf_allocated (Number of UD VBUFs allocated)	0	0	0	0	0	0
mv2_ud_vbuf_available (Number of UD VBUFs available)	0	0	0	0	0	0
mv2_ud_vbuf_freed (Number of UD VBUFs freed)	0	0	0	0	0	0
mv2_ud_vbuf_inuse (Number of UD VBUFs inuse)	0	0	0	0	0	0
mv2_ud_vbuf_max_use (Maximum number of UD VBUFs used)	0	0	0	0	0	0
mv2_vbuf_allocated (Number of VBUFs allocated)	320	320	320	0	1	320
mv2_vbuf_available (Number of VBUFs available)	255	255	255	0	1	255
mv2_vbuf_freed (Number of VBUFs freed)	25,545	25,545	25,545	0	1	25,545
mv2_vbuf_inuse (Number of VBUFs inuse)	65	65	65	0	1	65
mv2_vbuf_max_use (Maximum number of VBUFs used)	65	65	65	0	1	65
num_calloc_calls (Number of MPIT_calloc calls)	89	89	89	0	1	89

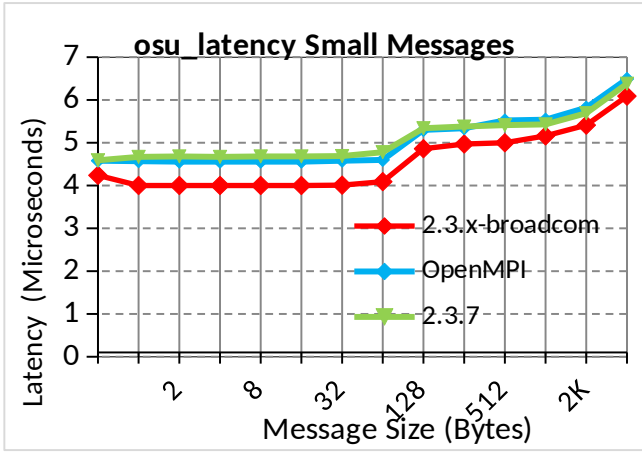
VBUF usage with CVAR based tuning as displayed by ParaProf

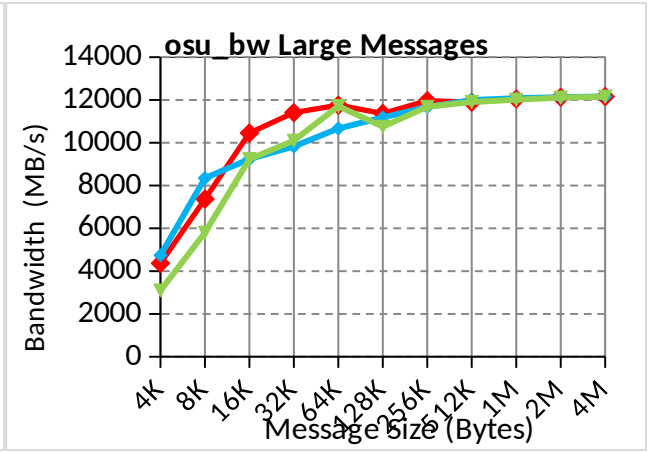
TAU: ParaProf: Context Events for: node 0 - bt-mz.E.vbuf_pool_16.1k.ppk

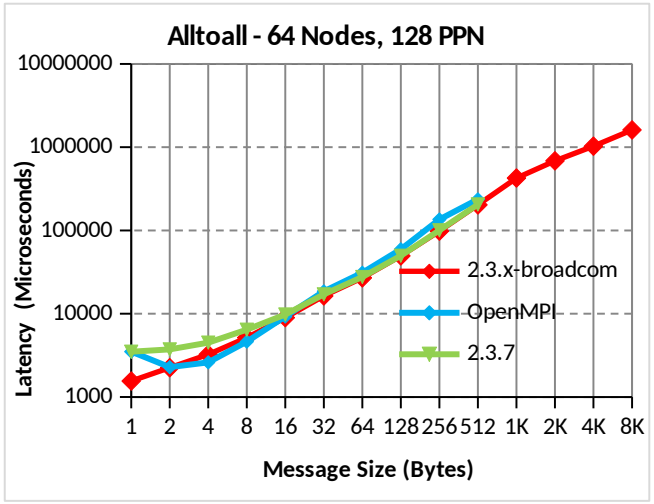
Name	MaxValue	MinValue	MeanValue	Std. Dev.	NumSamp...	Total
mv2_total_vbuf_memory (Total amount of memory in bytes used for VBUFs)	1,815,056	1,815,056	1,815,056	0	1	1,815,056
mv2_ud_vbuf_allocated (Number of UD VBUFs allocated)	0	0	0	0	0	0
mv2_ud_vbuf_available (Number of UD VBUFs available)	0	0	0	0	0	0
mv2_ud_vbuf_freed (Number of UD VBUFs freed)	0	0	0	0	0	0
mv2_ud_vbuf_inuse (Number of UD VBUFs inuse)	0	0	0	0	0	0
mv2_ud_vbuf_max_use (Maximum number of UD VBUFs used)	0	0	0	0	0	0
mv2_vbuf_allocated (Number of VBUFs allocated)	160	160	160	0	1	160
mv2_vbuf_available (Number of VBUFs available)	94	94	94	0	1	94
mv2_vbuf_freed (Number of VBUFs freed)	5,479	5,479	5,479	0	1	5,479
mv2_vbuf_inuse (Number of VBUFs inuse)	66	66	66	0	1	66

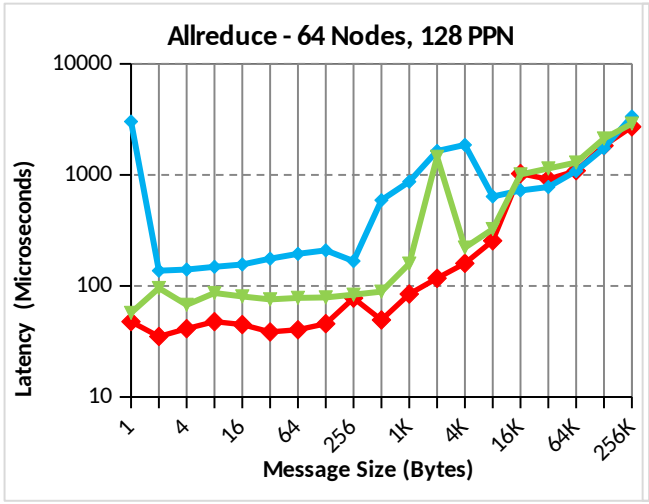
Performance Evaluation on Broadcom RoCE

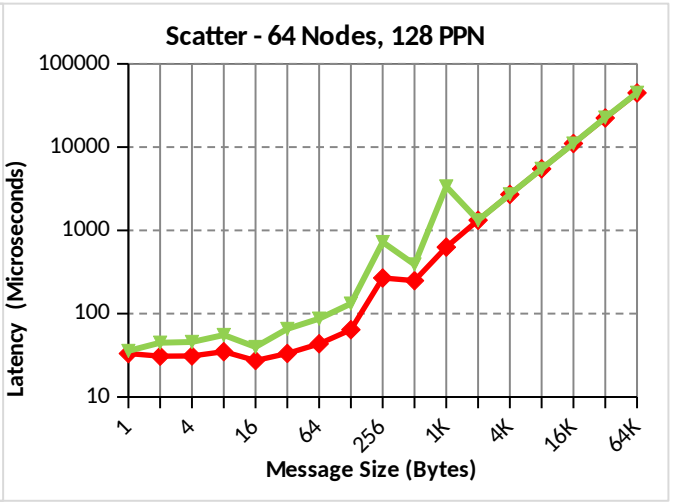
- Experimental results from Dell Bluebonnet
- Up to 20% reduction in small message point-to-point latency
- From 0.1x to 2x increase in bandwidth
- Up to 12.4x lower MPI_Allreduce latency
- Up to 5x lower MPI_Scatter latency



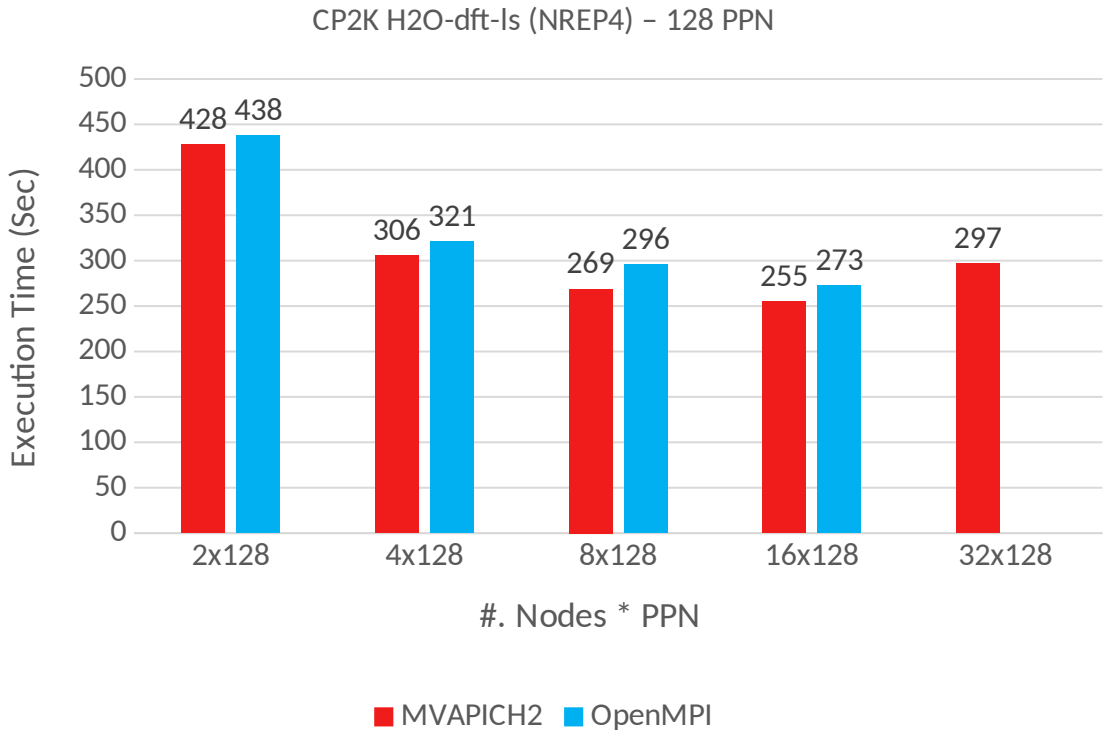


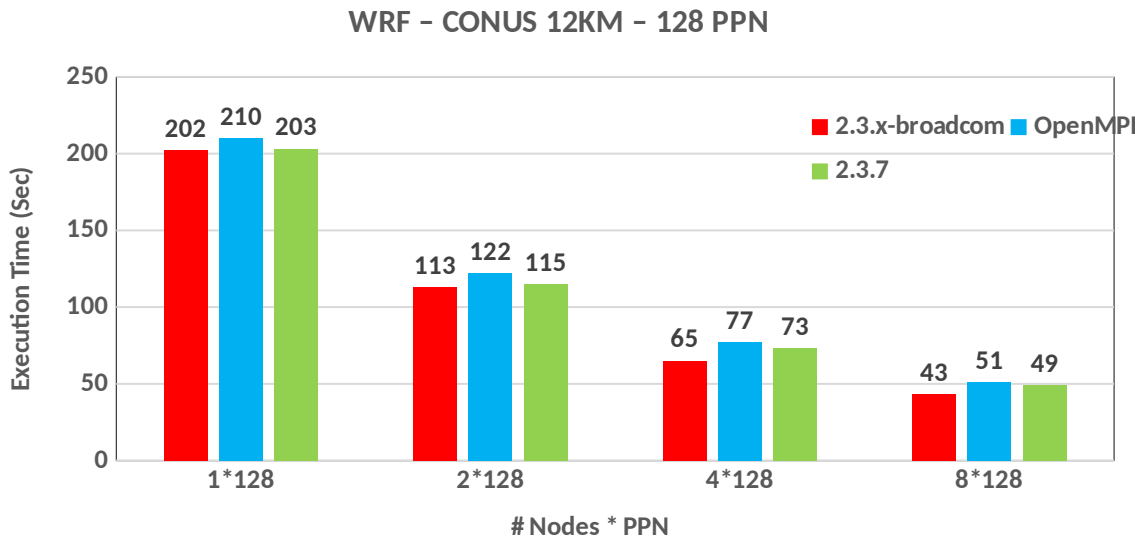


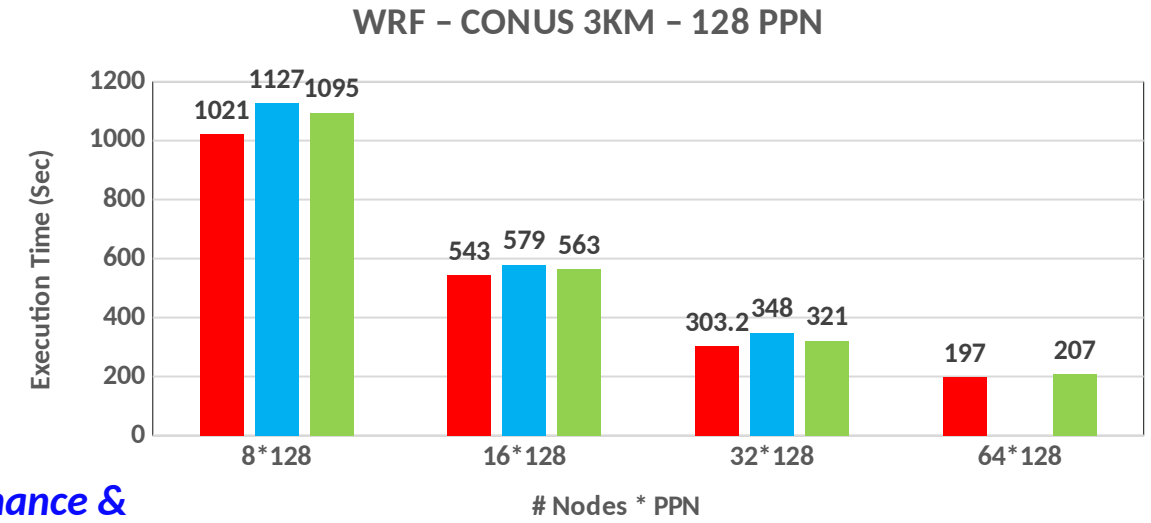




Performance Evaluation – Applications on Broadcom RoCE





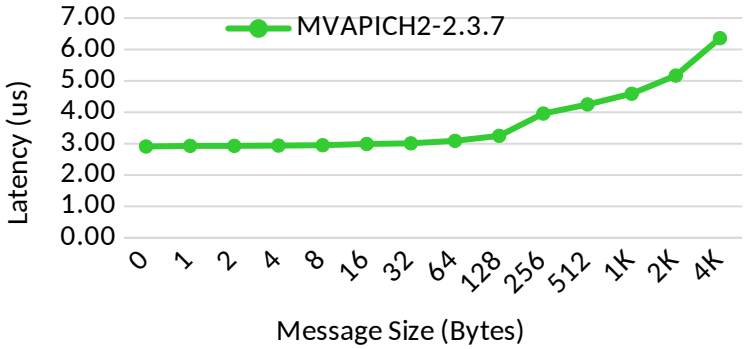


- Reduce up to 45% execution time of CP2K H2O-dft-ls (NREP4)
- Reduce up to 7% execution time of WRF CONUS 3KM

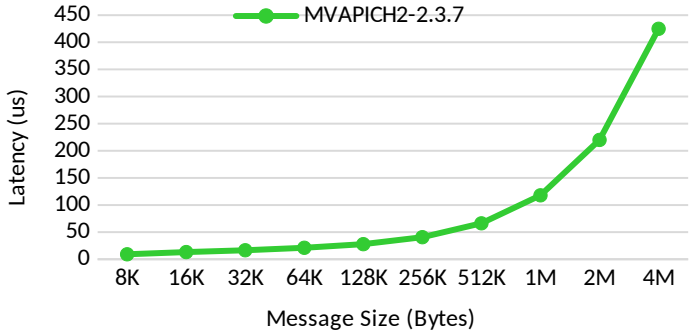
More information in yesterday's Short Talk by Shulei Xu titled "High Performance & Scalable MPI library over Broadcom RoCEv2" on August 21st , 4PM – 5:30PM

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

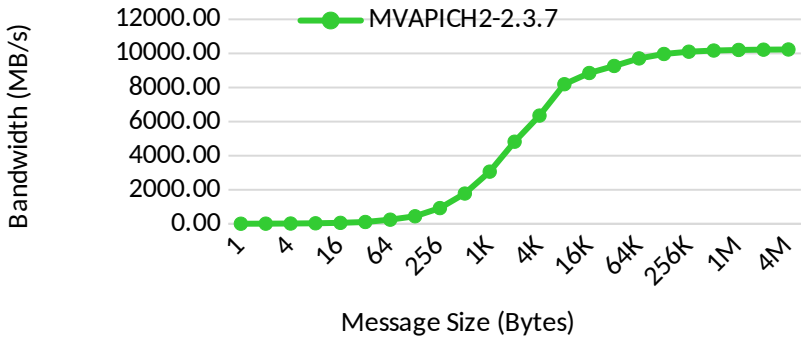
Small message Latency



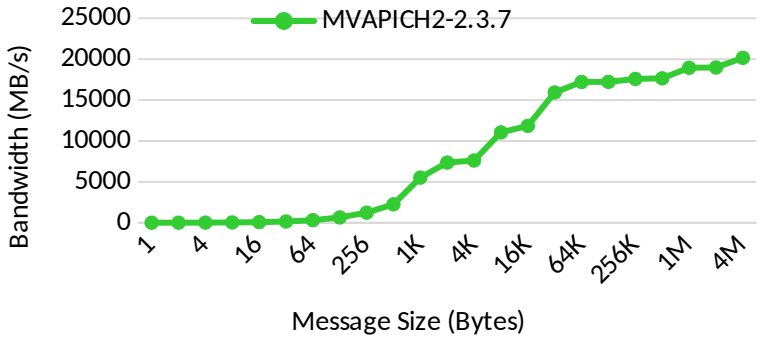
Medium/Large message Latency



Uni-directional Bandwidth



Bi-Directional Bandwidth



- Use multiple QPs to deliver performance

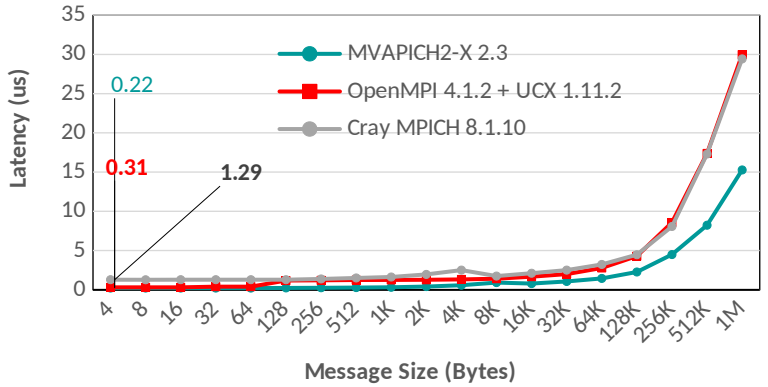
MVAPICH2 delivers

- 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**

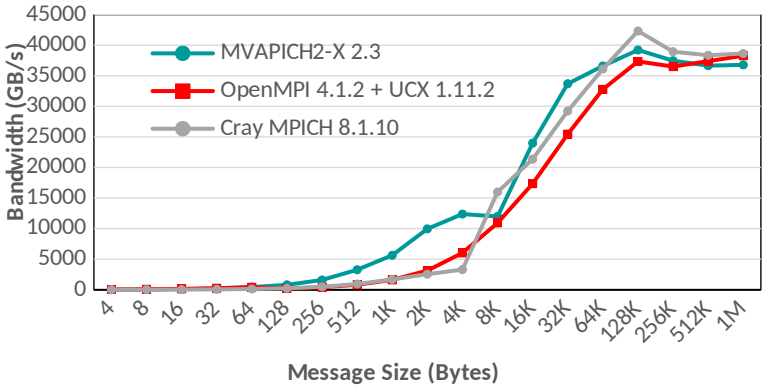
MVAPICH2 on Slingshot 10 - CPU

Point-to-Point – Intra-Node

Latency:

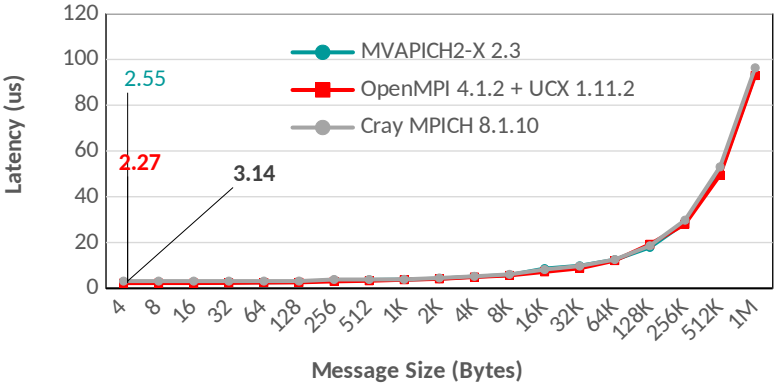


Bandwidth:

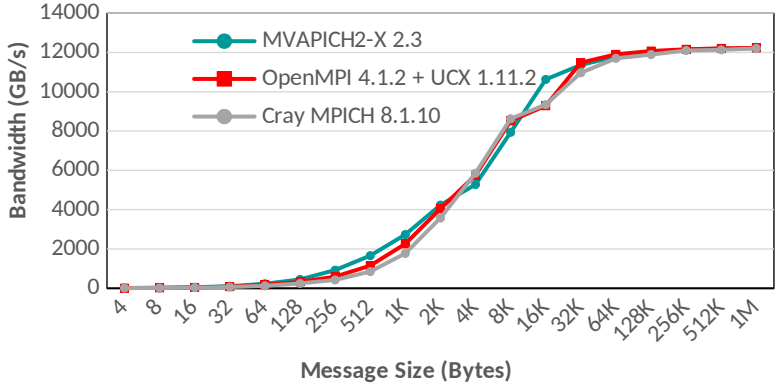


Point-to-Point – Inter-Node

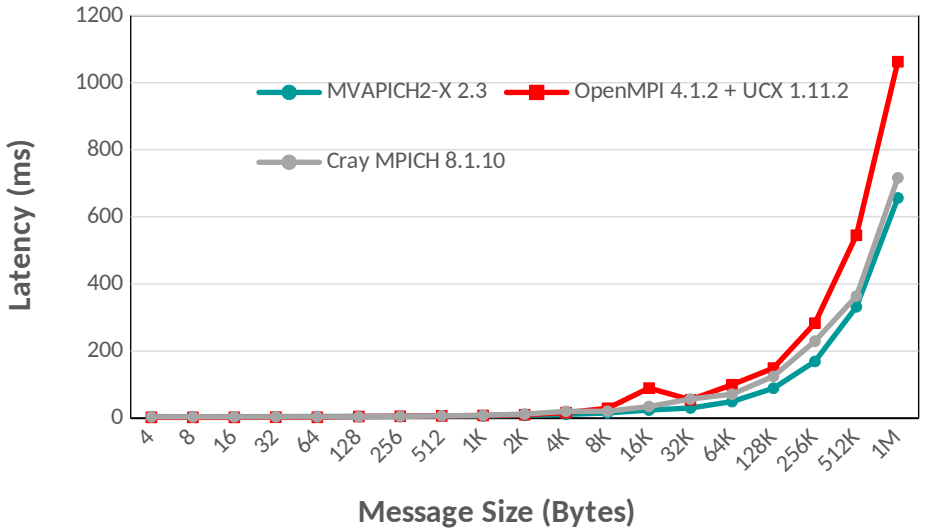
Latency:



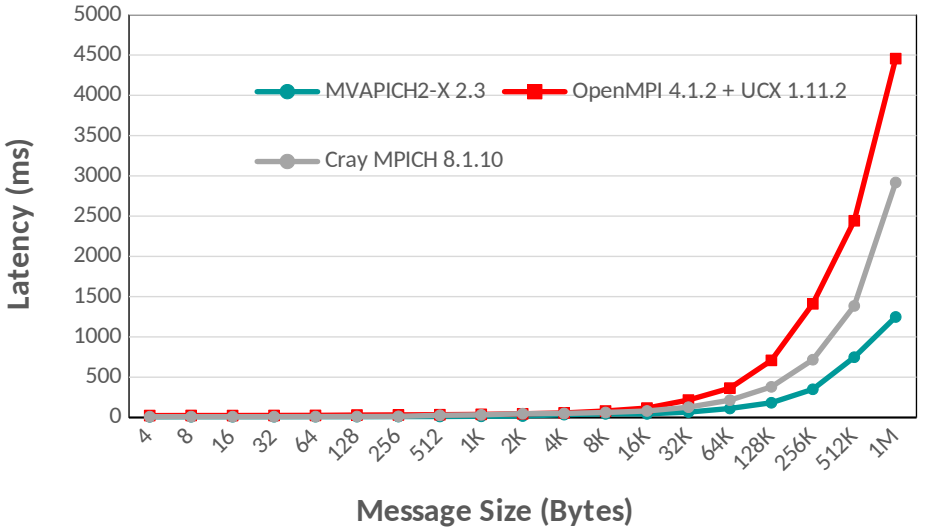
Bandwidth:



MPI_Bcast (4 Nodes, 64PPN)



MPI_Allreduce (4 Nodes, 64PPN)



AMD Epyc Rome CPUs

Outline

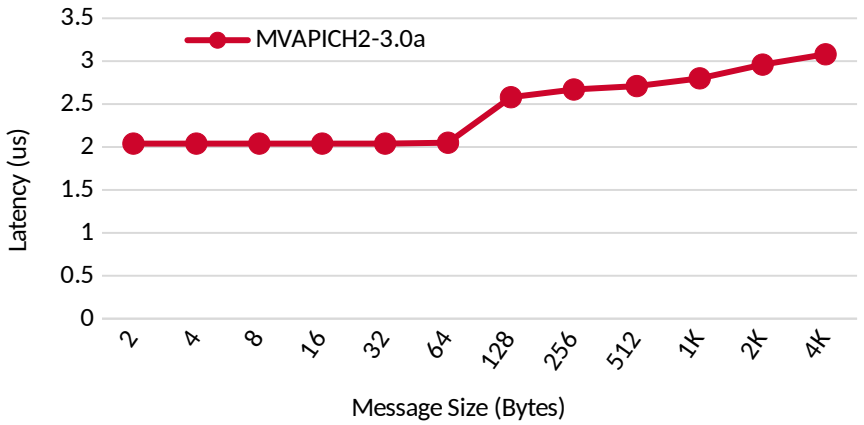
- Brief Overview of the MVAPICH Project
- **Features of Recent Releases**
  - MVAPICH2 2.3.7
  - **MVAPICH 3.0b**
  - MVAPICH2-J
  - MVAPICH2-X-AWS 2.3.7 and Cloud Deployments
  - MVAPICH2-GDR 2.3.7 for GPUs
  - MVAPICH-Plus 3.0a2
  - Converged software stack based on MVAPICH
    - Support for DL (HiDL), ML (MPI4cuML), Big Data (MPI4Spark), and Data Science (MPI4Dask)
  - OSU Micro-Benchmarks (OMB)
  - InfiniBand Network Analysis and Monitoring (INAM)
  - Applications: Best Practices
- Upcoming Features
- Conclusions

MVAPICH 3.0

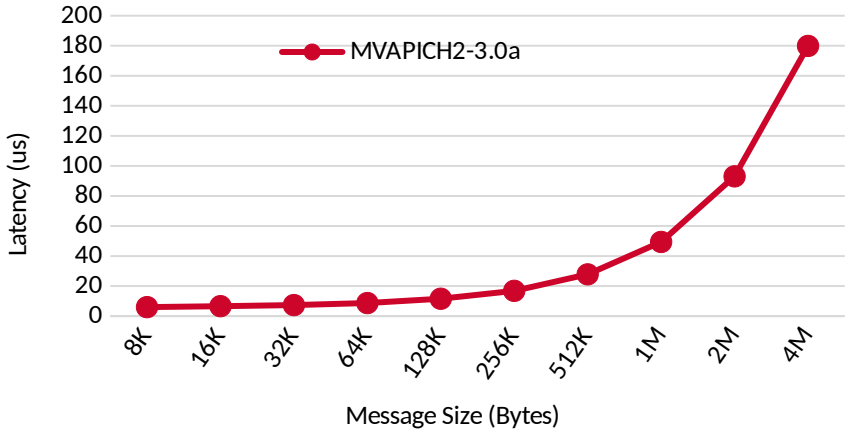
- **Beta version released 05/10/2023**
- Major Features and Enhancements
  - Based on MPICH 3.4.3
  - Support for OFI and UCX network libraries through MPICH netmod interface
  - MVAPICH enhanced collective designs
  - Support for Cray Slingshot 11 network
  - Improved CVAR interface for consistency and performance
  - Support for SHARP

MPI Level Latency on Slingshot 11

Small message Latency



Medium/Large message Latency

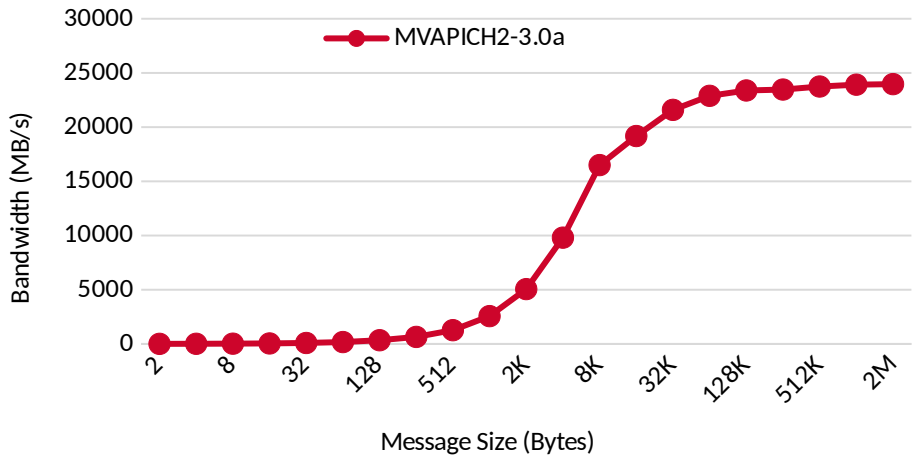


- **2us** inter-node point-to-point latency for small messages

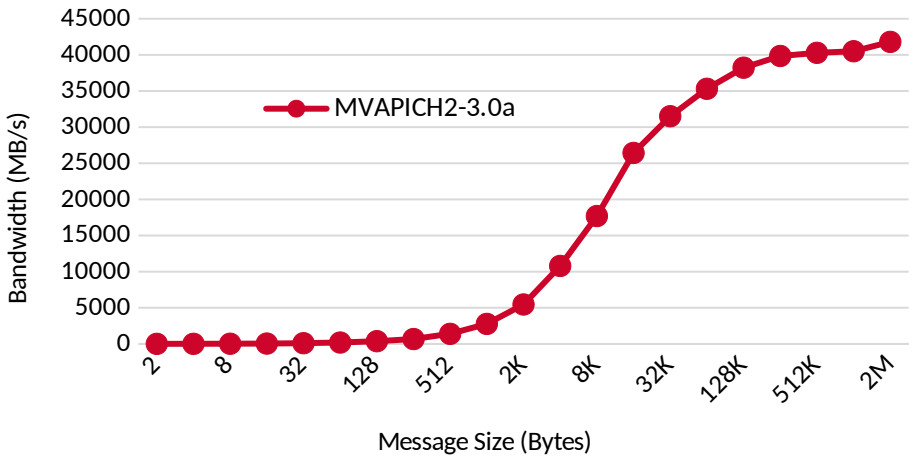
Interconnect : Cray HPE Slingshot 11
Library : MVAPICH2 3.0a
CPU : AMD EPYC 7763 (milan) Processor

MPI Level Bandwidth on Slingshot 11

Uni-directional Bandwidth



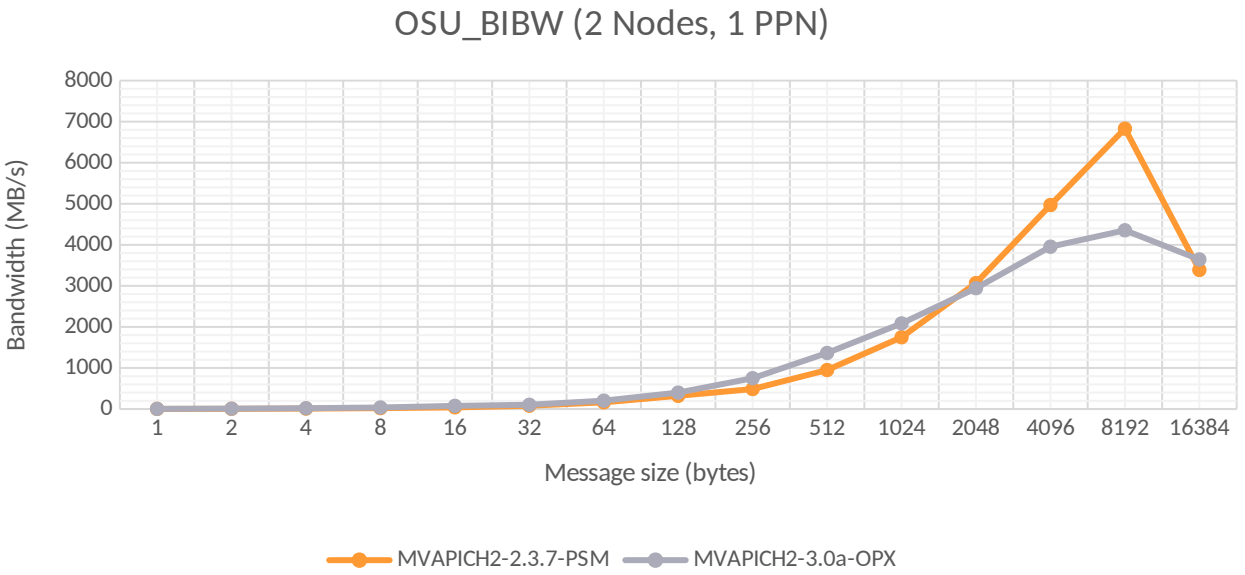
Bi-Directional Bandwidth

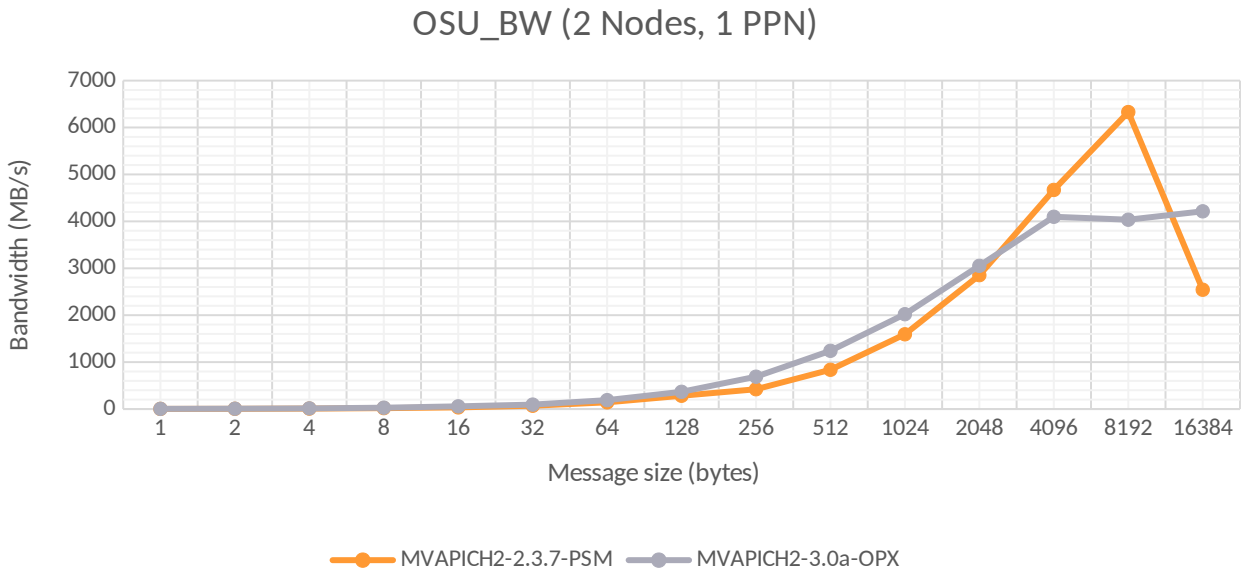


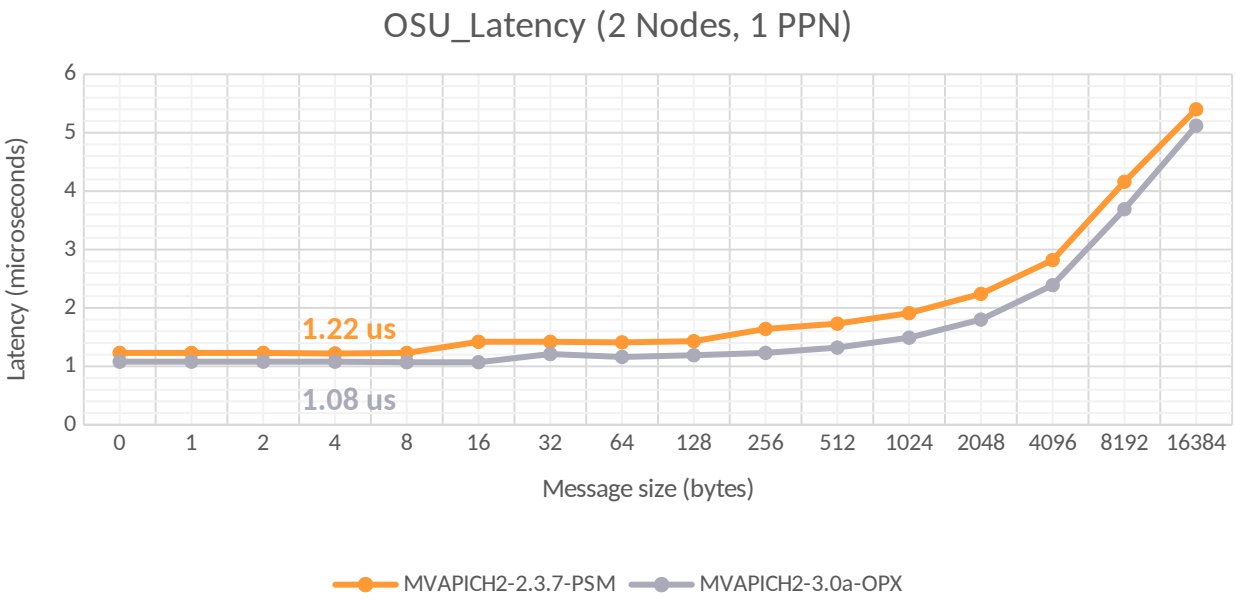
- **23,985 MB/s** uni-directional peak bandwidth
- **42,034 MB/s** bi-directional peak bandwidth

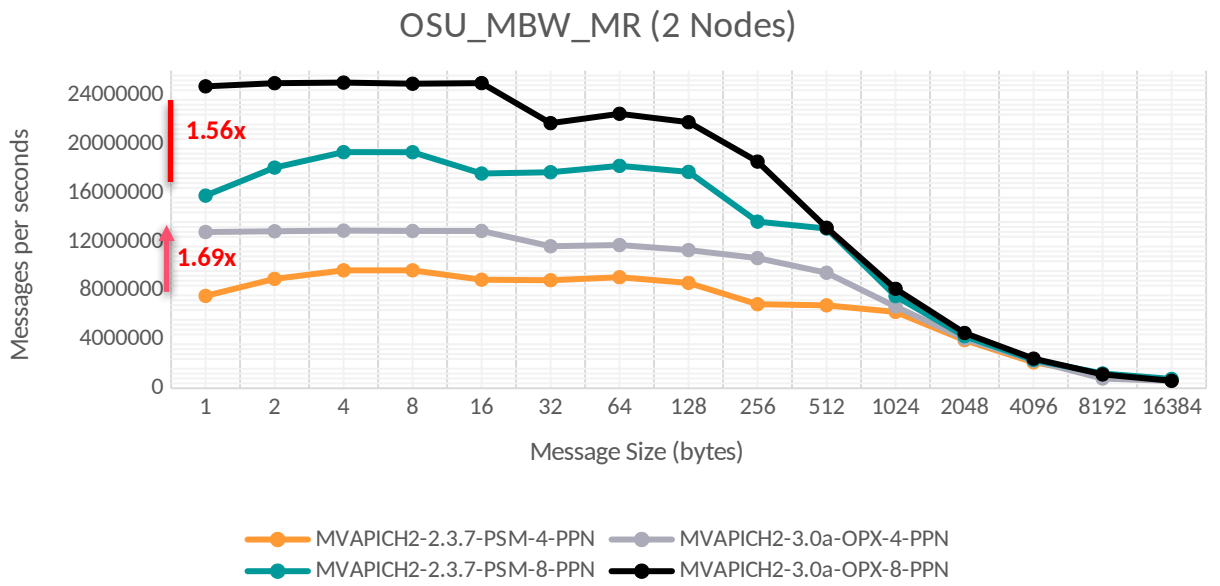
Interconnect : Cray HPE Slingshot 11 (200 Gbps)
Library : MVAPICH2 3.0a
CPU : AMD EPYC 7763 (milan) Processor

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







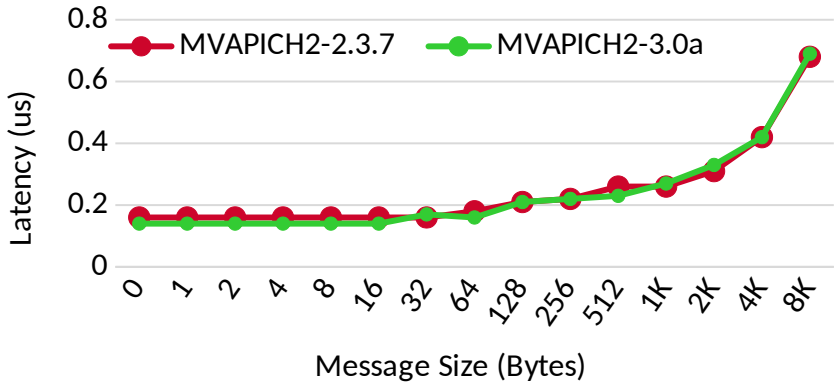


System: Intel Xeon Bronze (Skylake) 3106 CPU @ 1.70GHz (4 nodes, 16 cores/node, 8 x 2 sockets) with Omni-Path 100Gbps

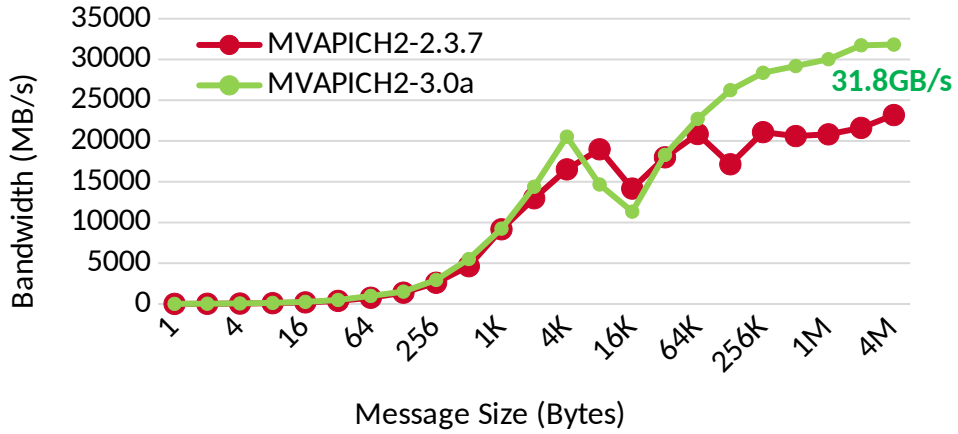
AMD Milan + HDR 200

Intra-Node CPU Point-to-Point

Latency

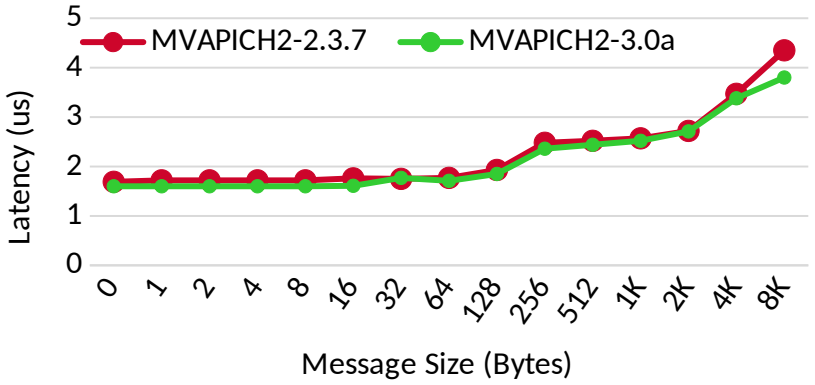


Bandwidth

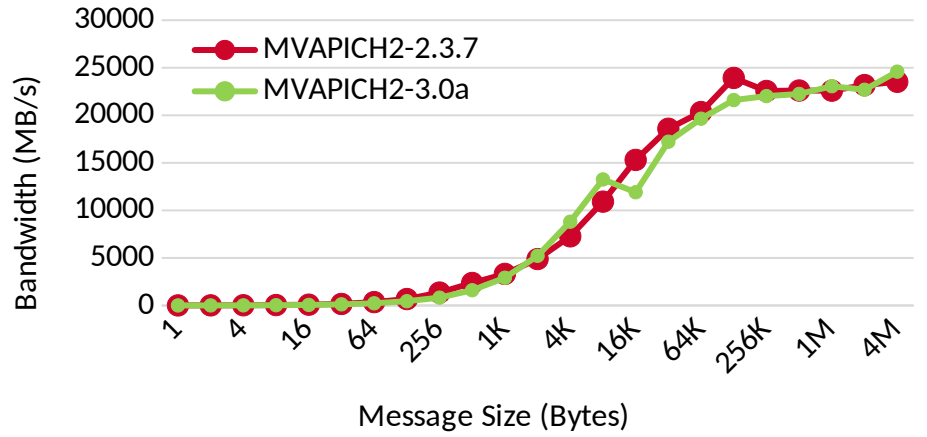


Inter-Node CPU Point-to-Point

Latency



Bandwidth



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

Outline

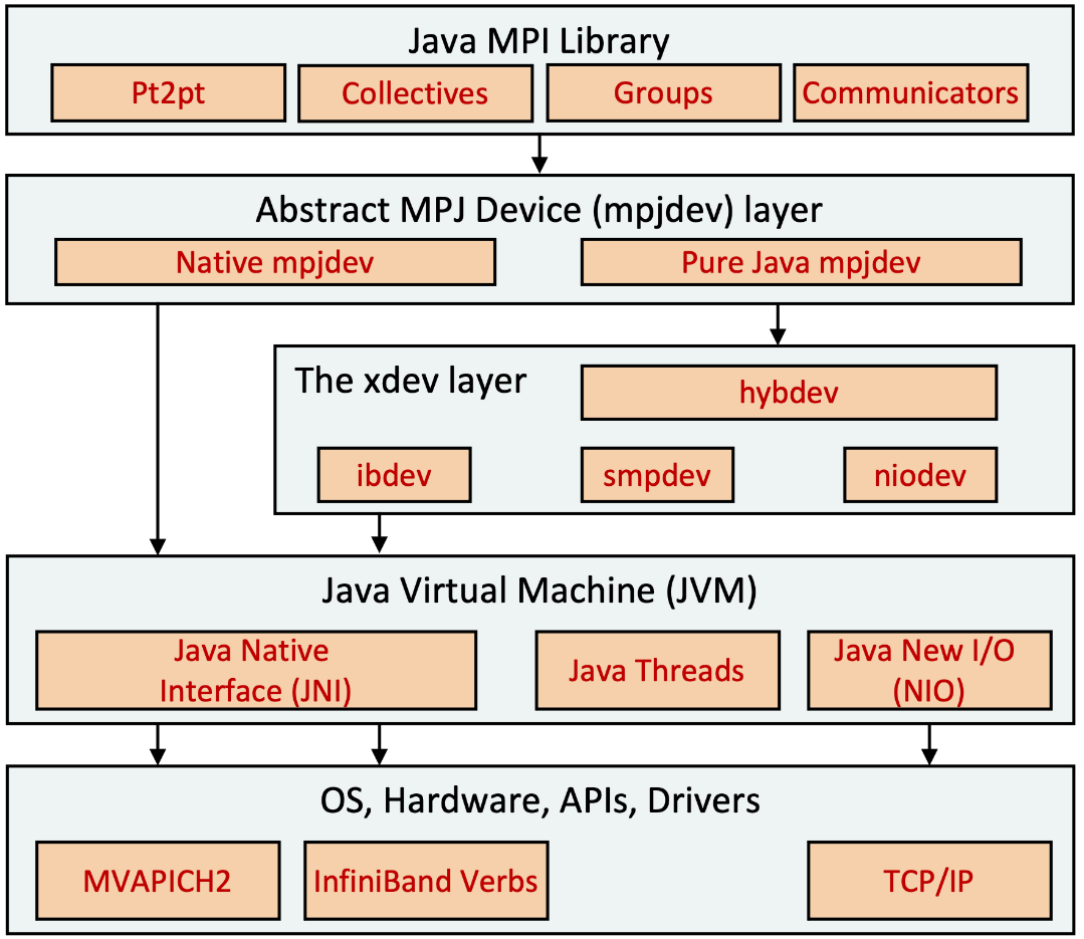
- Brief Overview of the MVAPICH Project
- **Features of Recent Releases**
  - MVAPICH2 2.3.7
  - MVAPICH 3.0b
  - **MVAPICH2-J**
  - MVAPICH2-X-AWS 2.3.7 and Cloud Deployments
  - MVAPICH2-GDR 2.3.7 for GPUs
  - MVAPICH-Plus 3.0a2
  - Converged software stack based on MVAPICH
    - Support for DL (HiDL), ML (MPI4cuML), Big Data (MPI4Spark), and Data Science (MPI4Dask)
  - OSU Micro-Benchmarks (OMB)
  - InfiniBand Network Analysis and Monitoring (INAM)
  - Applications: Best Practices
- Upcoming Features
- Conclusions

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.

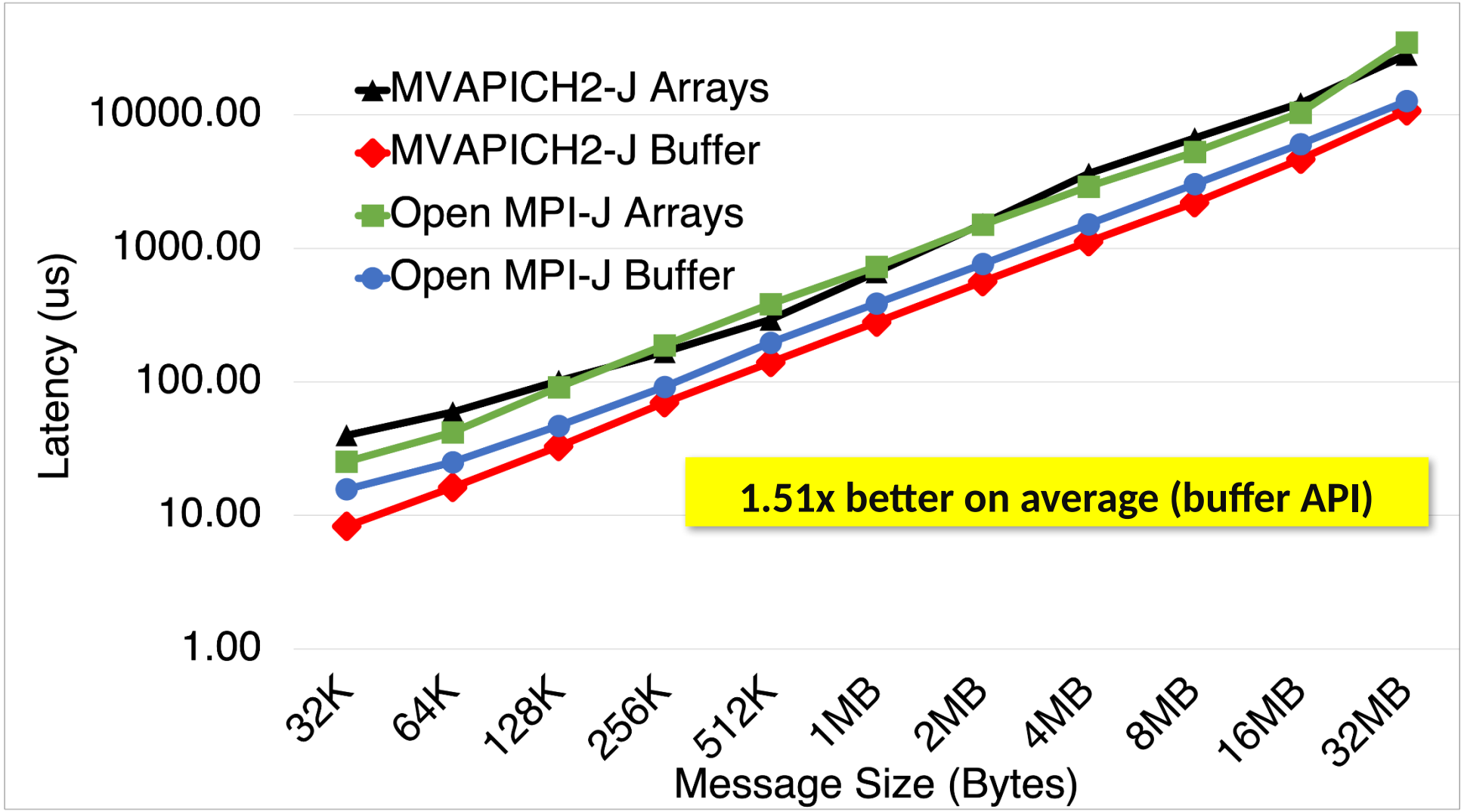
MVAPICH2-J: Java Bindings to MVAPICH2

- We have 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
- Used for Supporting Spark over MPI



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: Bcast Performance (8 processes)



Outline

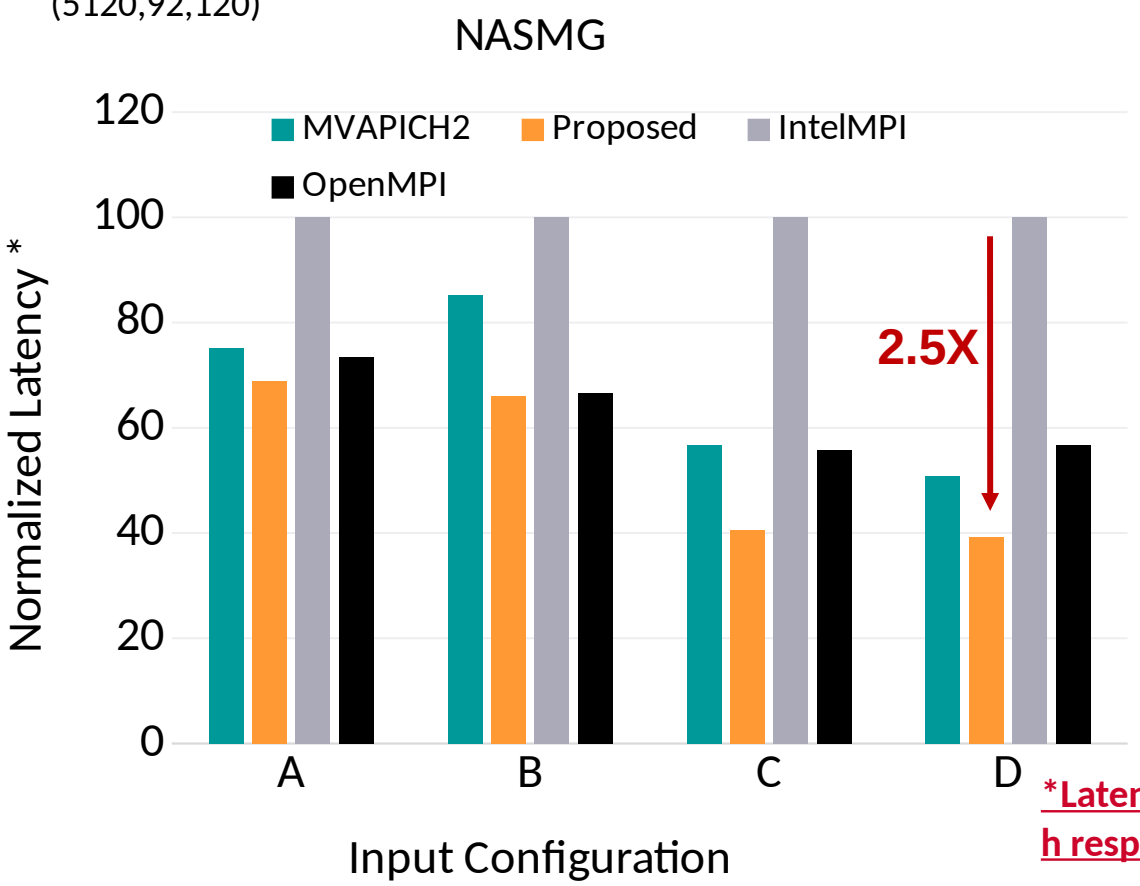
- Brief Overview of the MVAPICH Project
- **Features of Recent Releases**
  - MVAPICH2 2.3.7
  - MVAPICH 3.0b
  - MVAPICH2-J
  - **MVAPICH2-X-AWS 2.3.7 and Cloud Deployments**
  - MVAPICH2-GDR 2.3.7 for GPUs
  - MVAPICH-Plus 3.0a2
  - Converged software stack based on MVAPICH
    - Support for DL (HiDL), ML (MPI4cuML), Big Data (MPI4Spark), and Data Science (MPI4Dask)
  - OSU Micro-Benchmarks (OMB)
  - InfiniBand Network Analysis and Monitoring (INAM)
  - Applications: Best Practices
- Upcoming Features
- Conclusions

MVAPICH2 Software Family

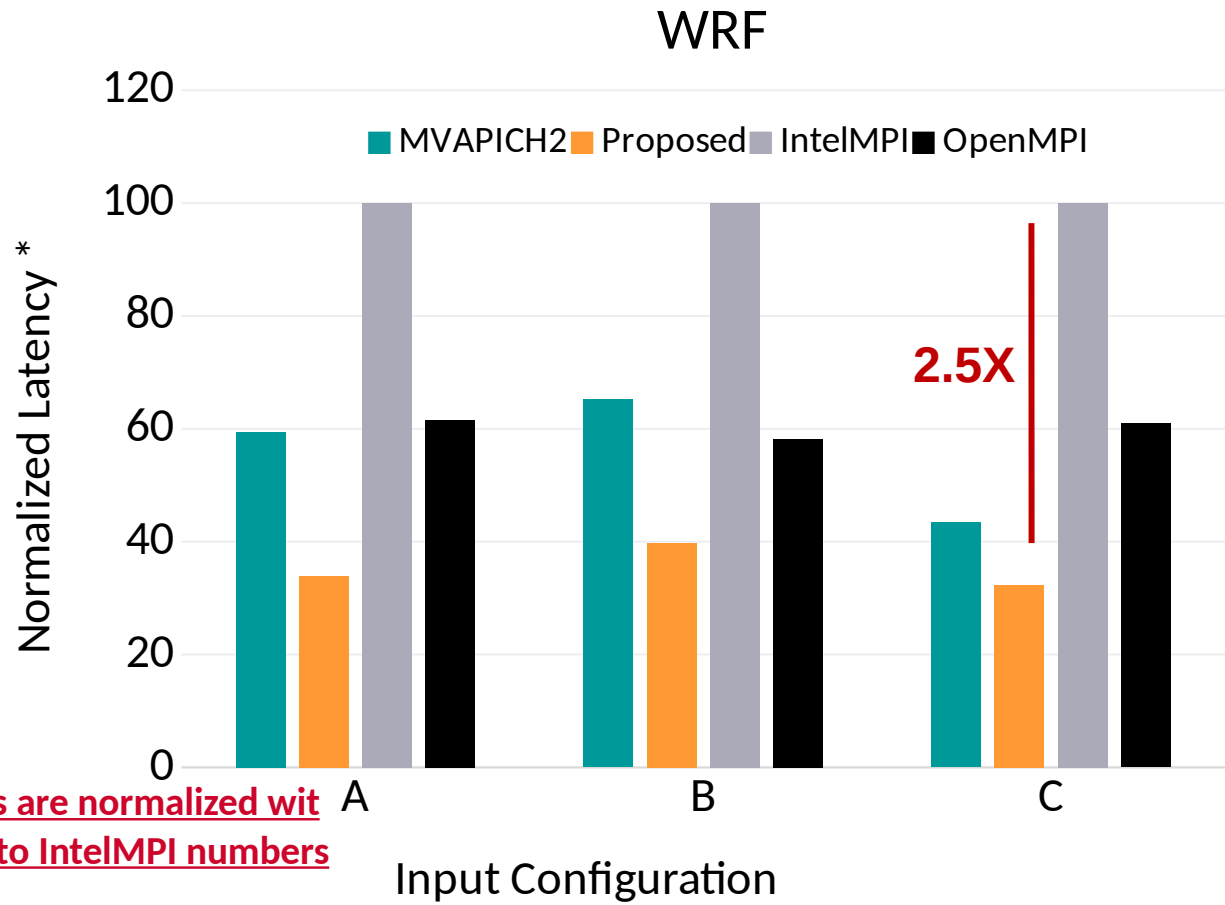
Requirements	Library
Advanced MPI features/support (UMR, ODP, DC, SHARP, XPMEM), OSU INAM (InfiniBand Network Monitoring and Analysis)	MVAPICH2-X
Advanced MPI features (SRD and XPMEM) with support for Amazon Elastic Fabric Adapter (EFA)	MVAPICH2-X-AWS

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

Collaboration with three Major Cloud Providers on MVAPICH2 Deployment

- Microsoft Azure
- AWS
- Oracle Cloud

MVAPICH2-Azure Deployment

- Azure used InfiniBand for its HPC instances
- All MVAPICH2 (and other libraries like HiBD and HiDL) can work in a transparent manner
- Integrated Azure CentOS HPC Images

  https://github.com/Azure/azhpc-images/releases/tag/centos-hpc-20210525

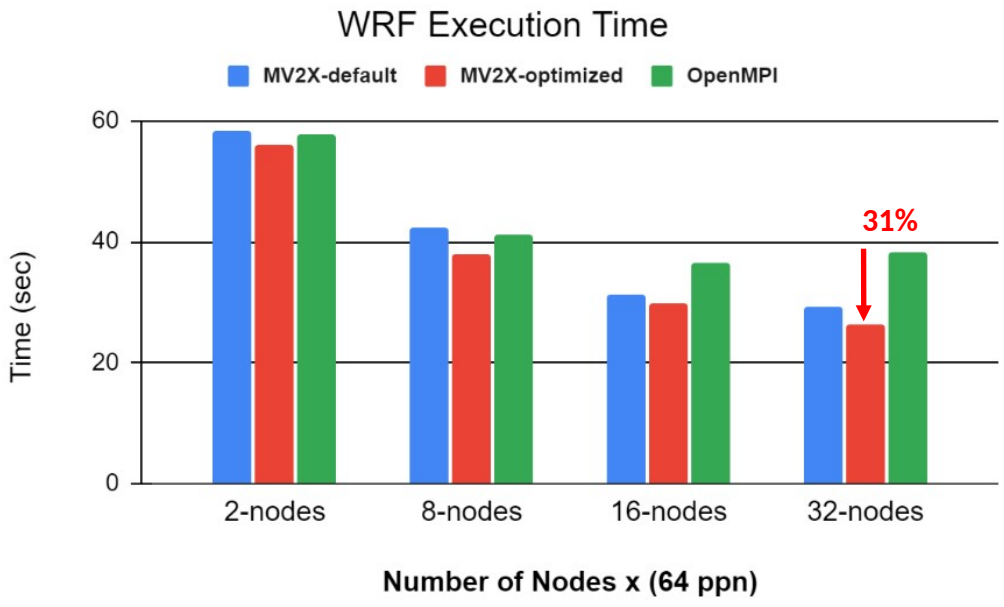
More on talk by Jithin Jose@Microsoft Azure today at 2:30-3:00 pm

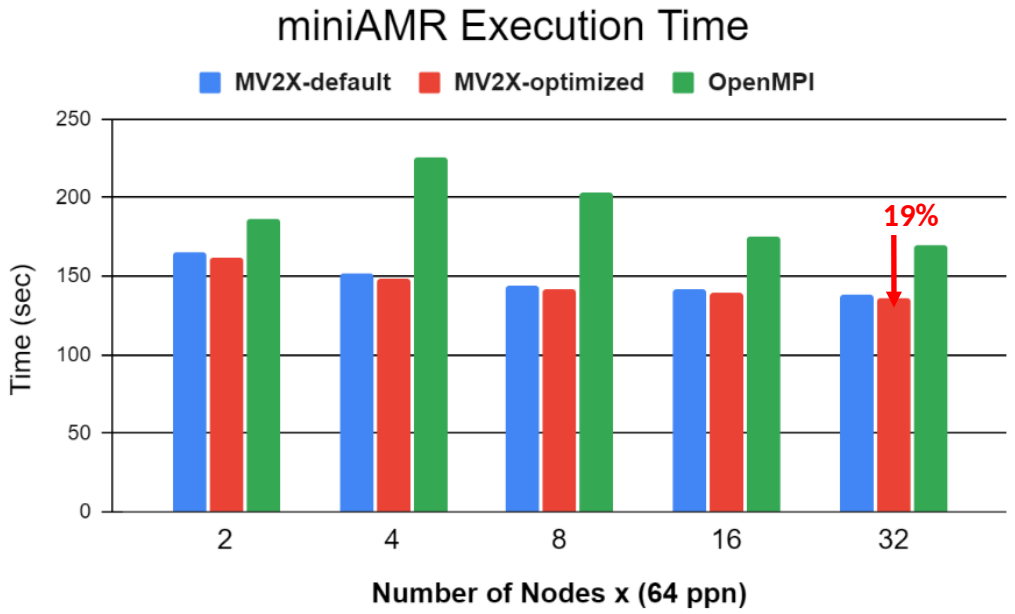
MVAPICH2-AWS Deployment

- AWS uses a proprietary Elastic Fabrics Adapter (EFA)
- Designed an optimized version of the MVAPICH2 library with EFA support
- MVAPICH2-X-AWS 2.3.7 release (08/21/22)
- Latest MVAPICH2 3.0a with Libfabric support can also work with EFA adapter (may not give the best performance)

Application Performance on Arm (Graviton) Instances

- Application-level performance comparison:
  - Weather Research Forecasting Application (WRF) with strong scaling input dataset from 12km resolution case over the Continental U.S. domain
  - miniAMR using default benchmarking input mesh size



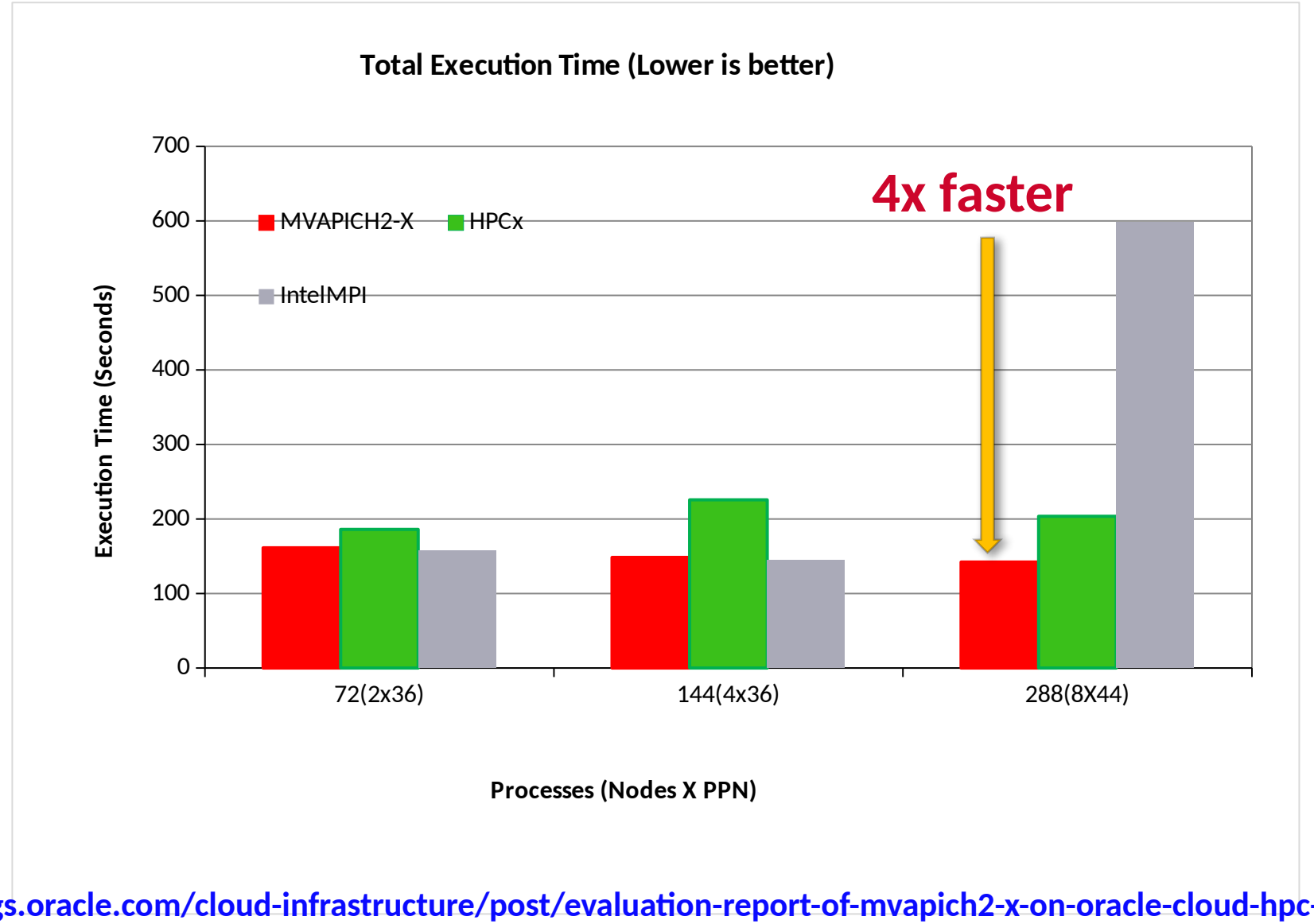


The Weather Research & Forecasting Model, https://www.mmm.ucar.edu/weather-research-and-forecasting-model
Adaptive Mesh Refinement Mini-App, https://github.com/Mantevo/miniAMR

MVAPICH2-Oracle Cloud Deployment

- Oracle cloud uses RoCE for its HPC instances
- All MVAPICH2 (and other libraries like HiBD and HiDL) can work in a transparent manner

MiniAMR Performance on Oracle Cloud



https://blogs.oracle.com/cloud-infrastructure/post/evaluation-report-of-mvapich2-x-on-oracle-cloud-hpc-shapes

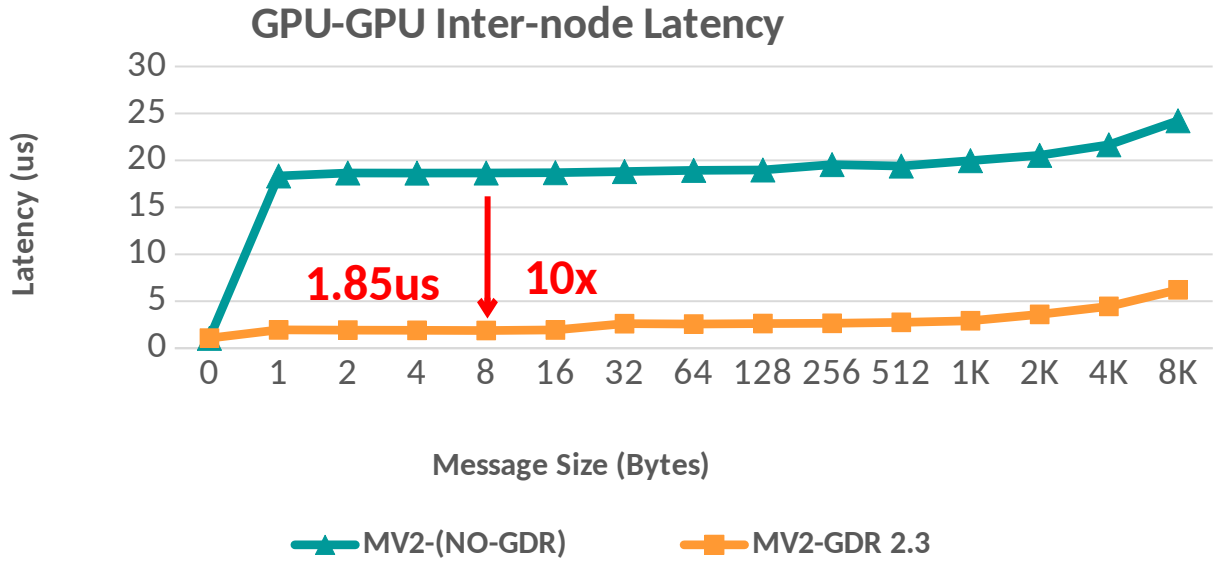
Outline

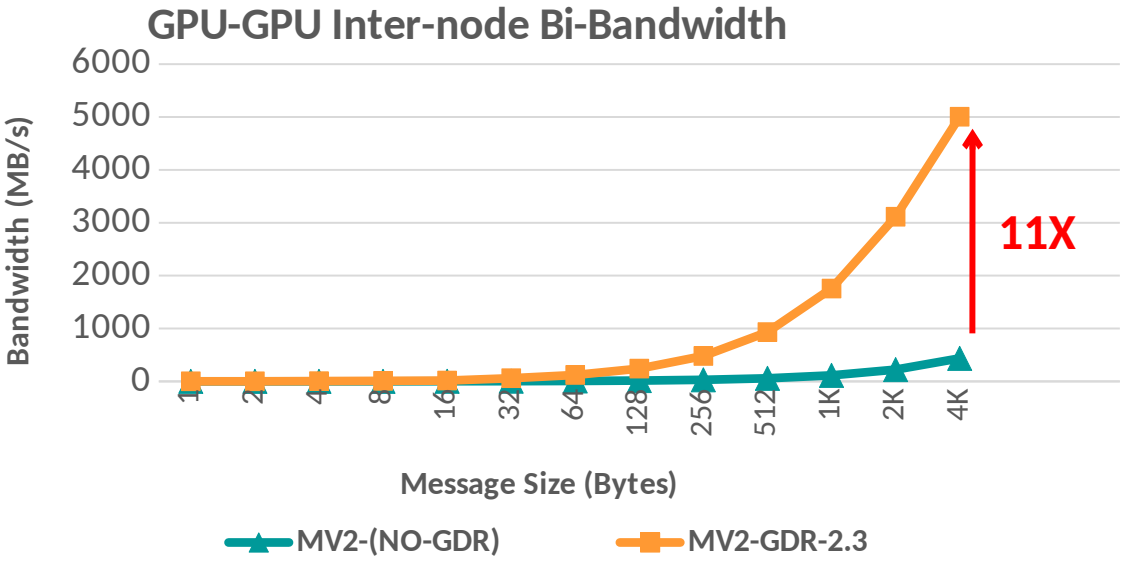
- Brief Overview of the MVAPICH Project
- **Features of Recent Releases**
  - MVAPICH2 2.3.7
  - MVAPICH 3.0b
  - MVAPICH2-J
  - MVAPICH2-X-AWS 2.3.7 and Cloud Deployments
  - **MVAPICH2-GDR 2.3.7 for GPUs**
  - MVAPICH-Plus 3.0a2
  - Converged software stack based on MVAPICH
    - Support for DL (HiDL), ML (MPI4cuML), Big Data (MPI4Spark), and Data Science (MPI4Dask)
  - OSU Micro-Benchmarks (OMB)
  - InfiniBand Network Analysis and Monitoring (INAM)
  - Applications: Best Practices
- Upcoming Features
- Conclusions

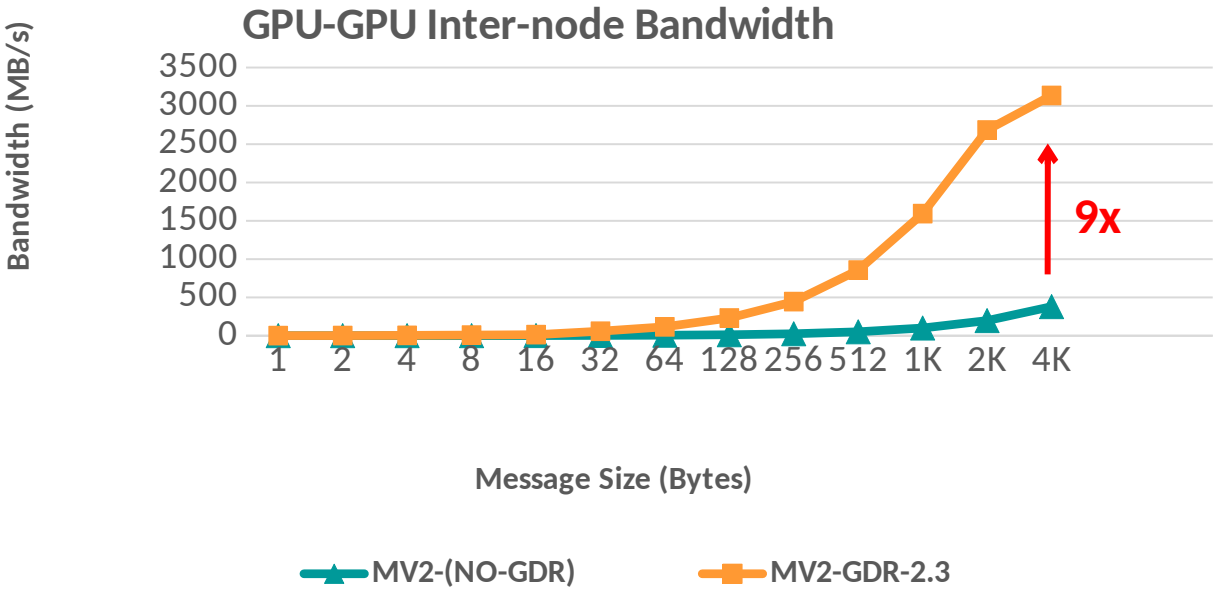
Highlights of some MVAPICH2-GDR Features

- CUDA-Aware MPI
- Derived Data Type (DDT) Support
- On-the-fly Compression for GPU-GPU Communication
- Optimized Collective Support for DGX-A100
- Support for
  - NVIDIA GPUs
  - AMD GPUs
  - Intel GPUs (is being worked out)
  - Slingshot 10 + AMD GPUs

Optimized MVAPICH2-GDR Design



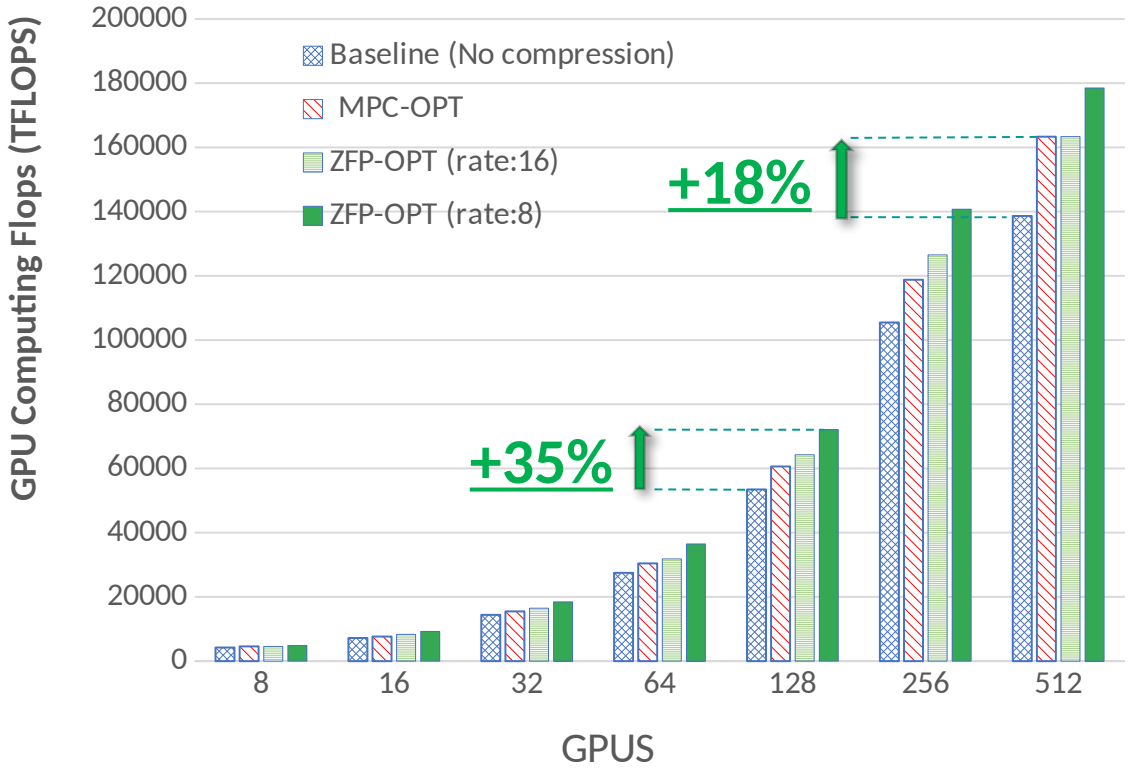


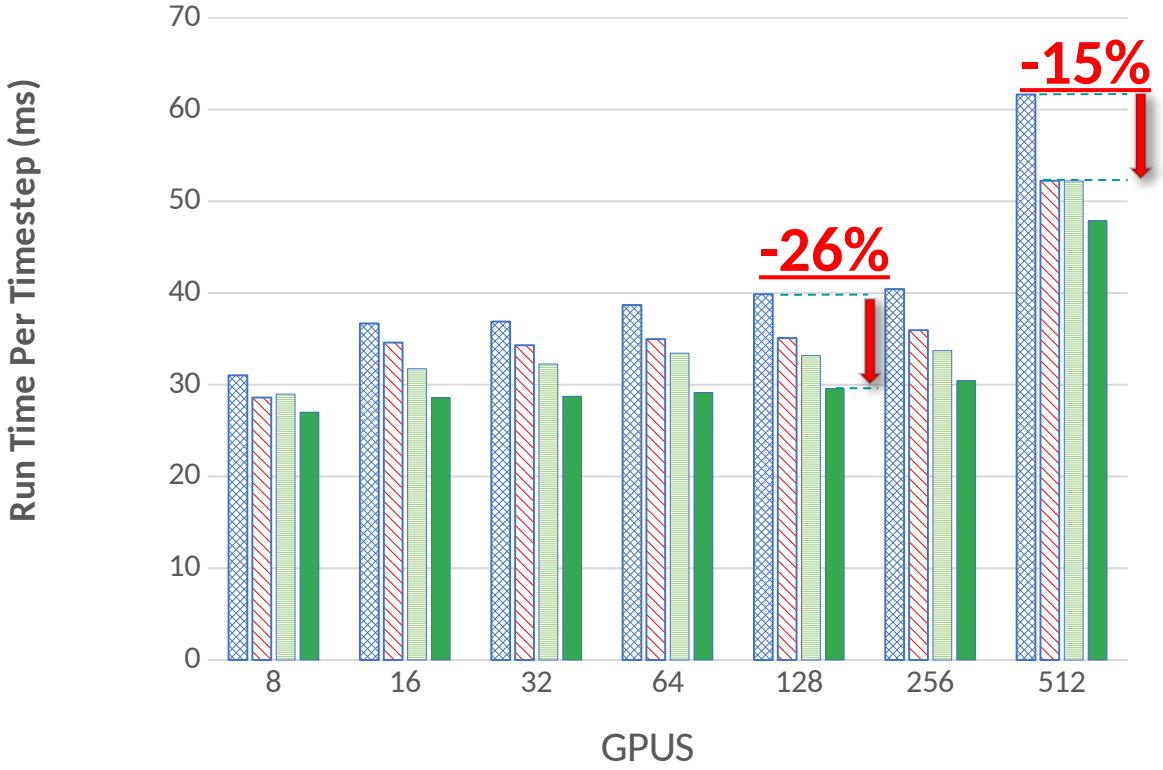


MVAPICH2-GDR-2.3.1
Intel Haswell (E5-2687W @ 3.10 GHz) node - 20 cores
NVIDIA Volta V100 GPU
Mellanox Connect-X4 EDR HCA
CUDA 9.0
Mellanox OFED 4.0 with GPU-Direct-RDMA

"On-the-fly" Compression Support in MVAPICH2-GDR

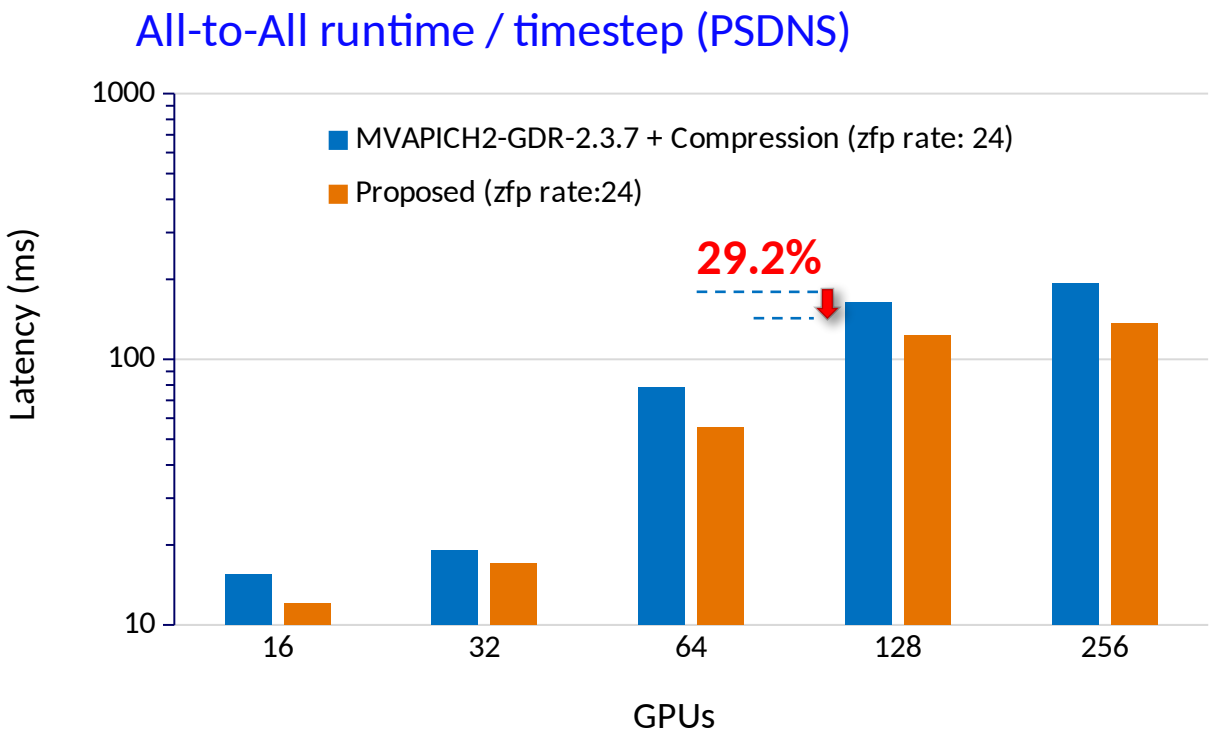
- Weak-Scaling of HPC application **AWP-ODC** on Lassen cluster (V100 nodes)
- MPC-OPT achieves up to **+18%** GPU computing flops, **-15%** runtime per timestep
- ZFP-OPT achieves up to **+35%** GPU computing flops, **-26%** runtime per timestep

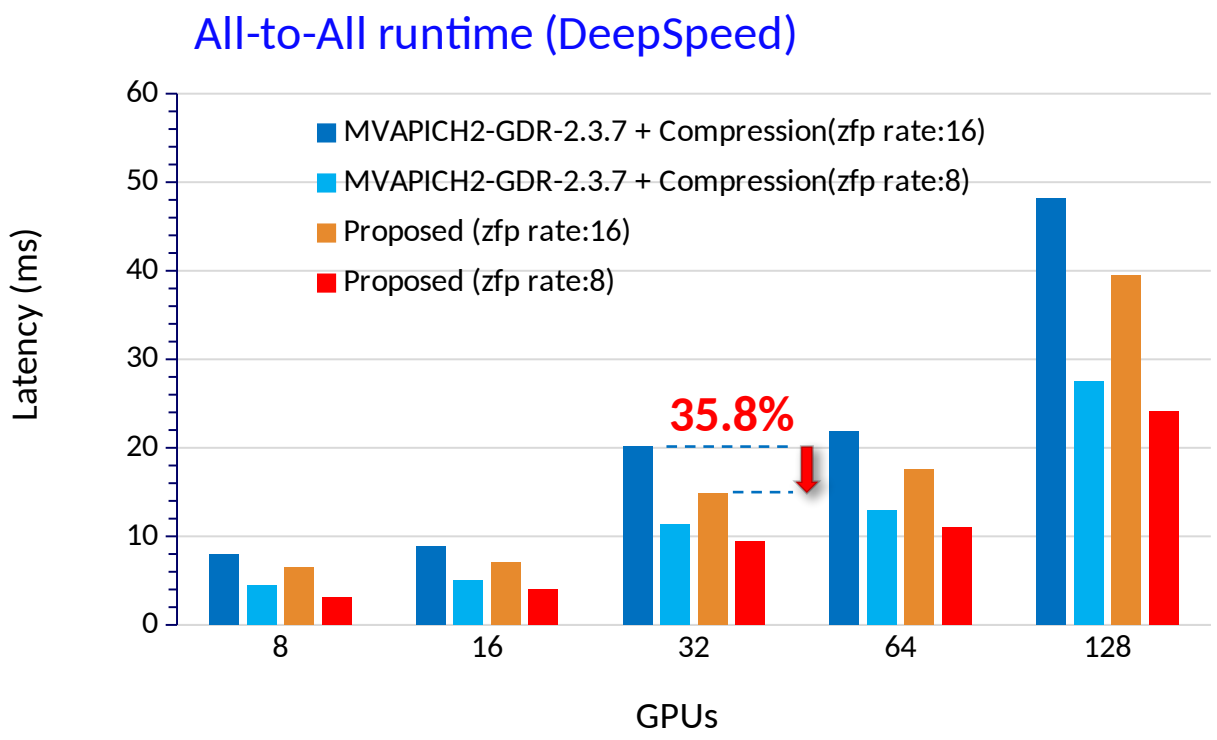




Q. Zhou, C. Chu, N. Senthil Kumar, P. Kousha, M. Ghazimirsaeed, H. Subramoni, and D.K. Panda, Designing High-Performance MPI Libraries with On-the-fly Compression for Modern GPU Clusters, 35th IEEE International Parallel & Distributed Processing Symposium (IPDPS), May 2021. [Best Paper Finalist]

Performance of All-to-All with Online Compression





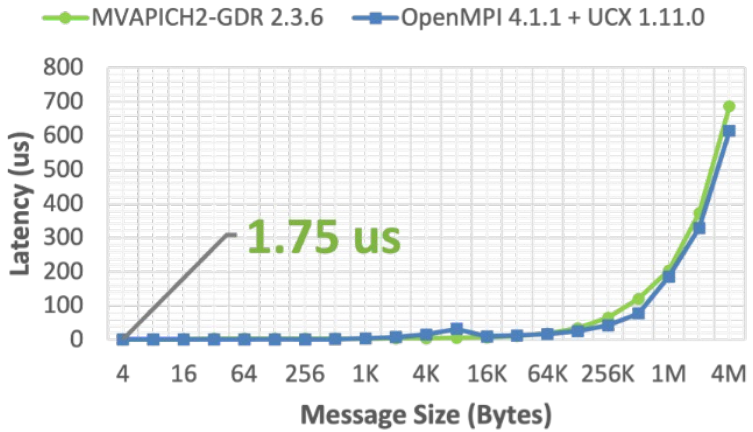
- Improvement compared to MVAPICH2-GDR-2.3.7 with Point-to-Point compression
  - 3D-FFT: Reduce All-to-All runtime by up to 29.2% with ZFP(rate: 24) on 64 GPUs
  - DeepSpeed benchmark: Reduce All-to-All runtime by up to 35.8% with ZFP(rate: 16) on 32 GPUs

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.

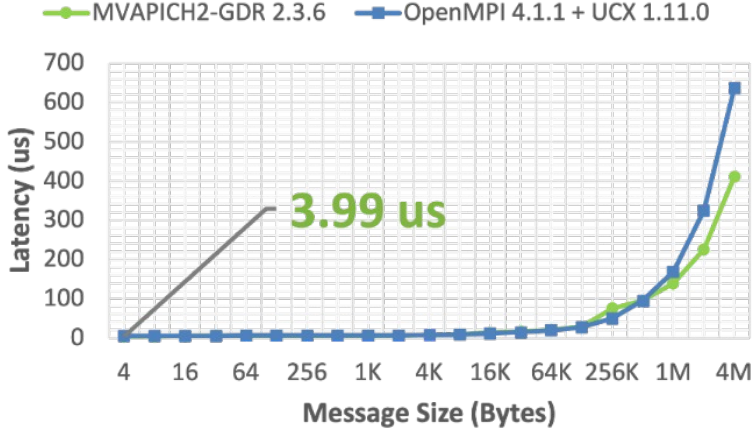
Available in MVAPICH2-GDR 2.3.7

ROCm-aware MVAPICH2-GDR - Support for AMD GPUs

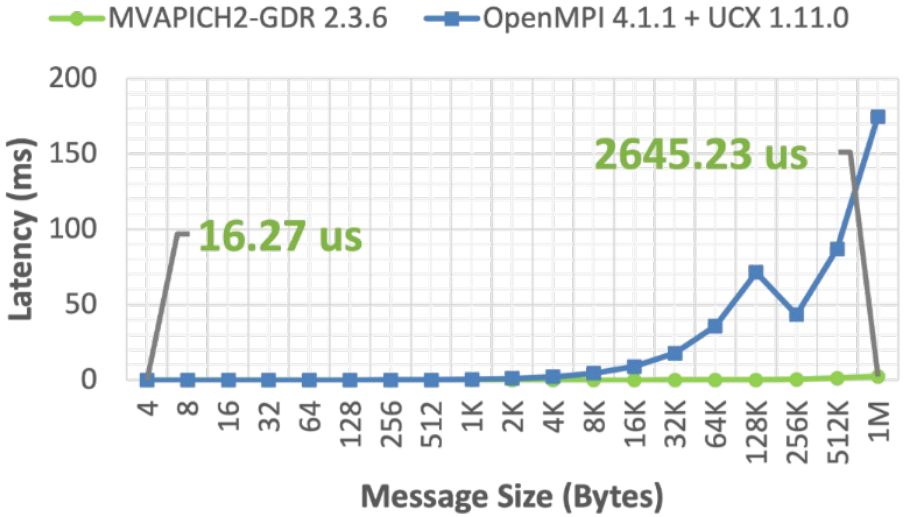
Intra-Node Point-to-Point Latency



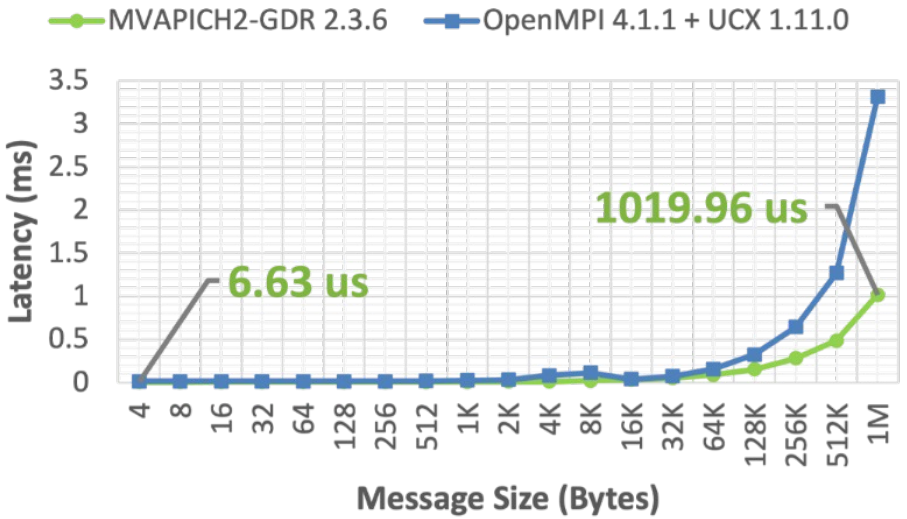
Inter-Node Point-to-Point Latency



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)

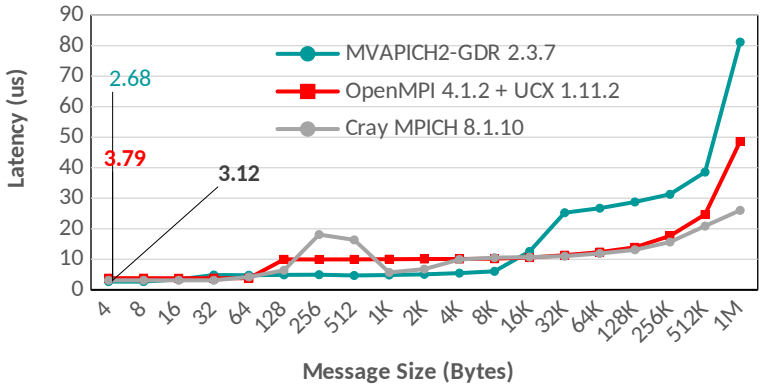
Available since MVAPICH2-GDR 2.3.5+ & OMB v5.7+

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.

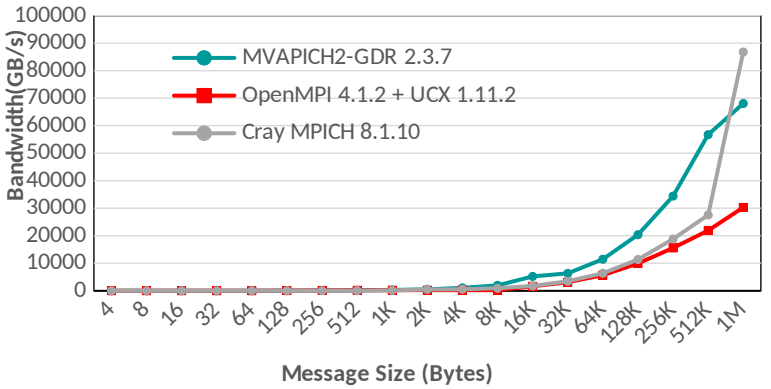
MVAPICH2-GDR on Slingshot-10 - GPU

Point-to-Point – Intra-Node

Latency:

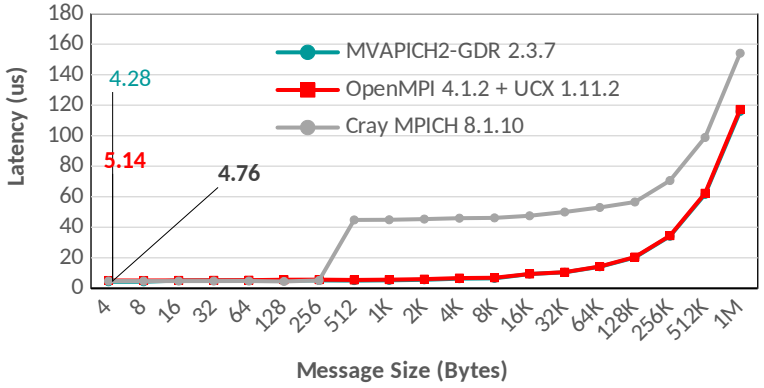


Bandwidth:

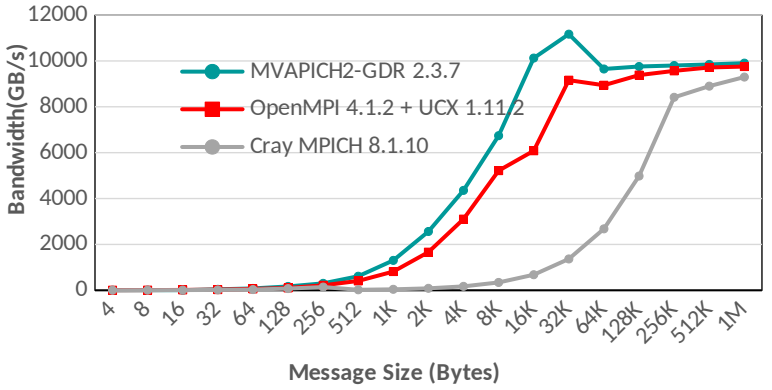


Point-to-Point – Inter-Node

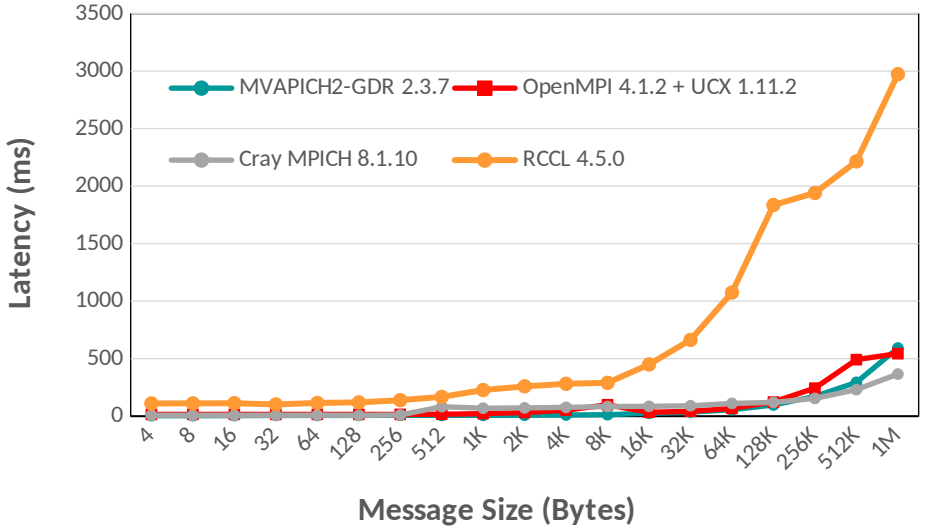
Latency:



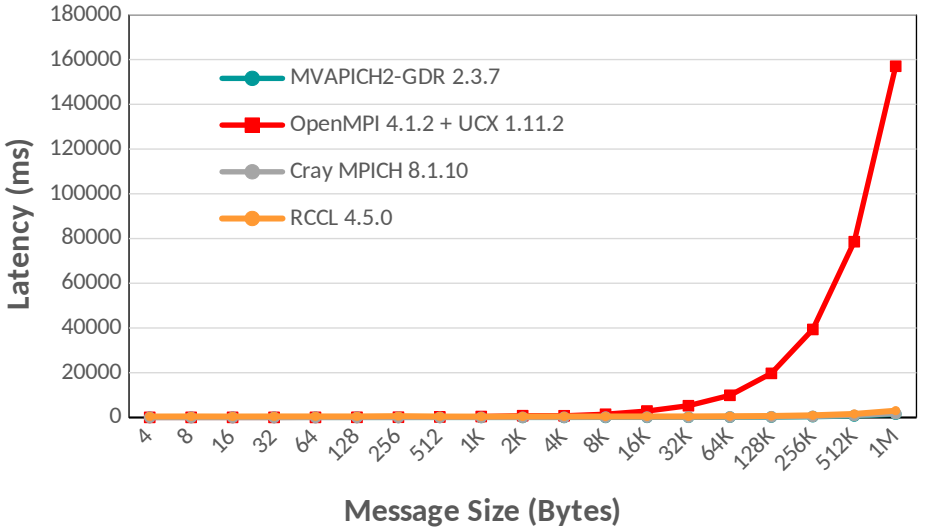
Bandwidth:



MPI_Bcast (4 Nodes, 64PPN)



MPI_Allreduce (4 Nodes, 64PPN)

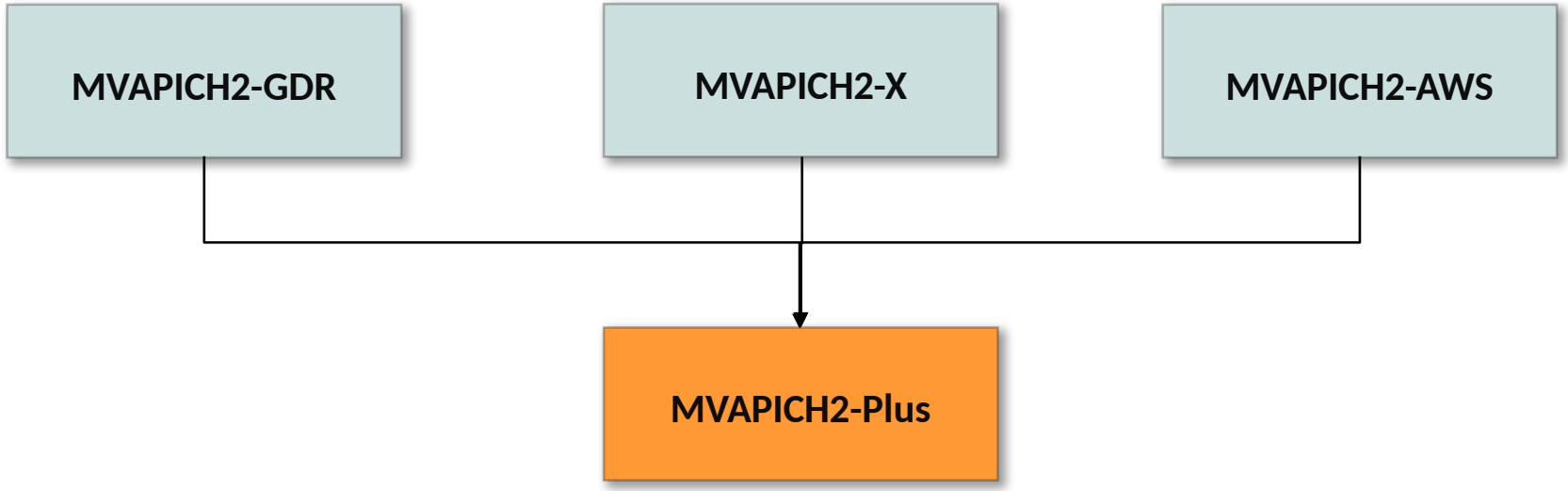


AMD Epyc Rome CPUs and AMD MI100 GPUs

Outline

- Brief Overview of the MVAPICH Project
- **Features of Recent Releases**
  - MVAPICH2 2.3.7
  - MVAPICH 3.0b
  - MVAPICH2-J
  - MVAPICH2-X-AWS 2.3.7 and Cloud Deployments
  - MVAPICH2-GDR 2.3.7 for GPUs
  - **MVAPICH-Plus 3.0a2**
  - Converged software stack based on MVAPICH
    - Support for DL (HiDL), ML (MPI4cuML), Big Data (MPI4Spark), and Data Science (MPI4Dask)
  - OSU Micro-Benchmarks (OMB)
  - InfiniBand Network Analysis and Monitoring (INAM)
  - Applications: Best Practices
- Upcoming Features
- Conclusions

MVAPICH2-Plus: One Stack for all Architectures and Interconnects



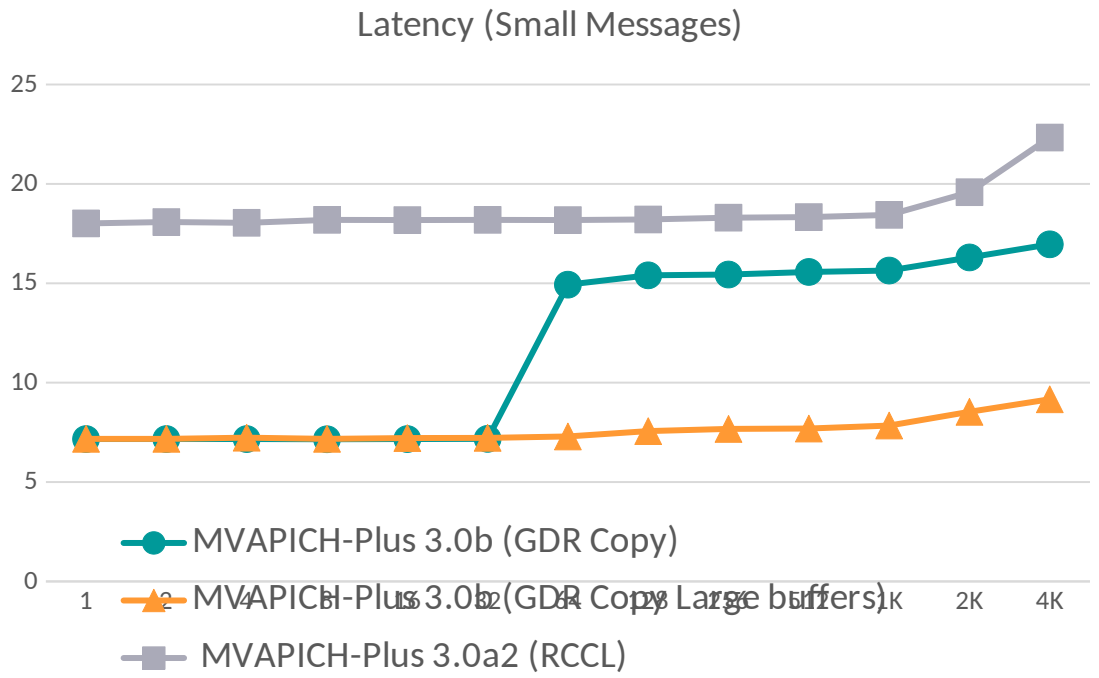
- Various libraries are being merged into one release
- Simplifies installation and deployment
- Enables utilizing the best advanced features for all architectures

MVAPICH-Plus 3.0

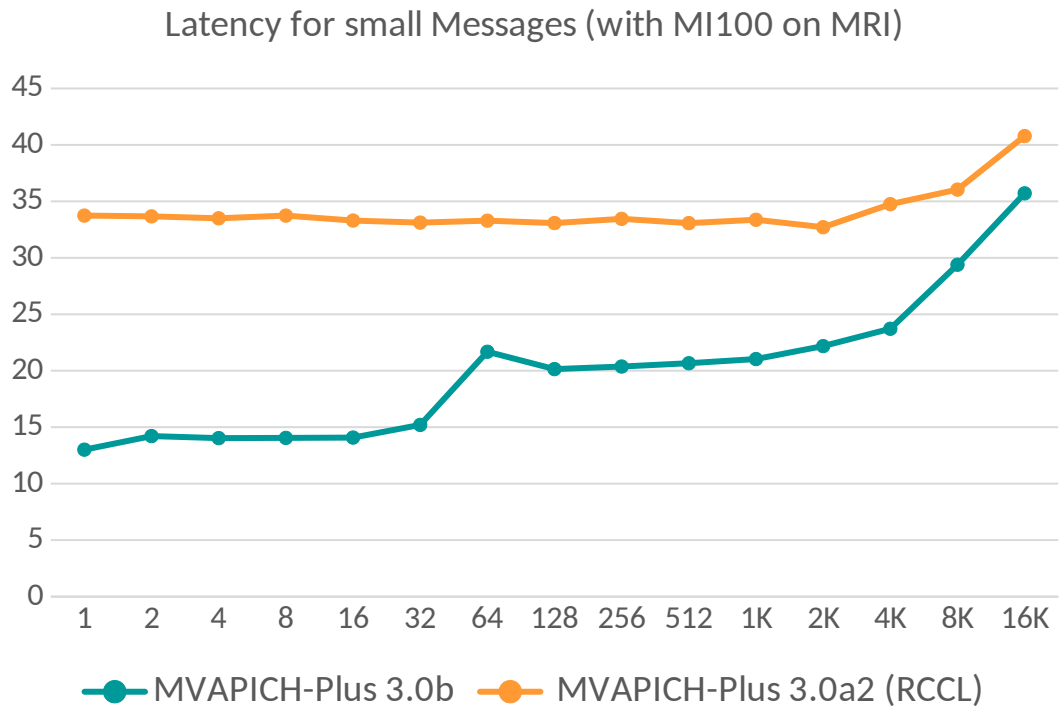
- Latest Alpha2 Release 07/19/2023
- Major Features and Enhancements
  - Based on MVAPICH 3.0
  - Supports all networks types supported by
  - Support for AMD and NVIDIA GPUs
  - Support for MVAPICH enhanced GPU Collectives
- Upcoming Beta Release
  - Support for enhanced GPU pt2pt operations
    - GDRCOPY and IPC

MVAPICH-PLUS with GDRCopy Design on AMD GPUs

Latency:



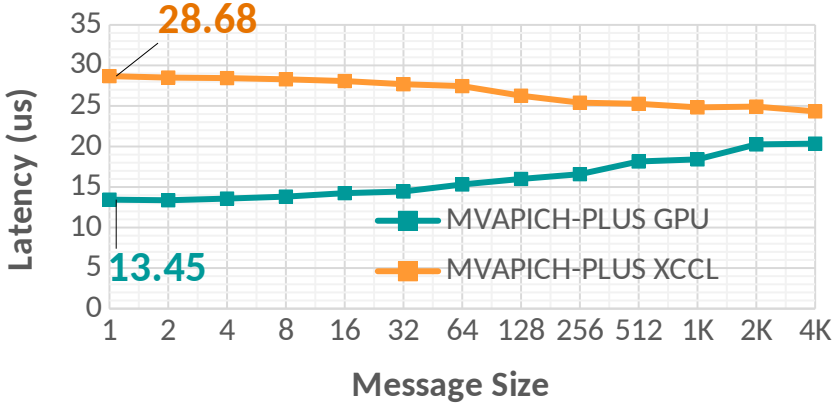
TIOGA - AMD MI100 GPUs

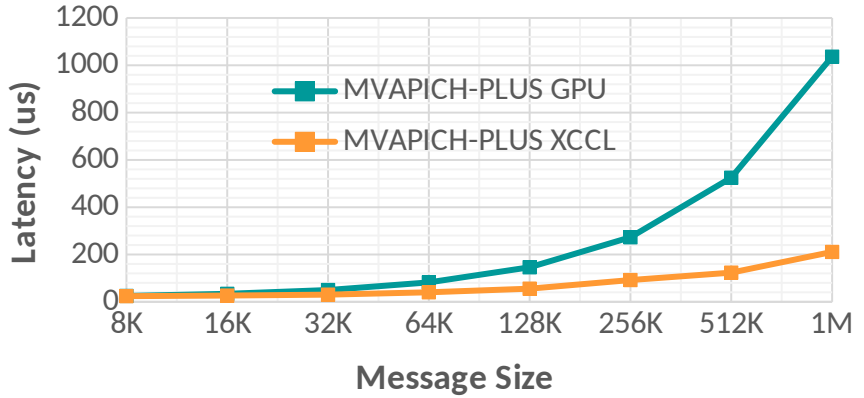


AMD GPUs on MRI

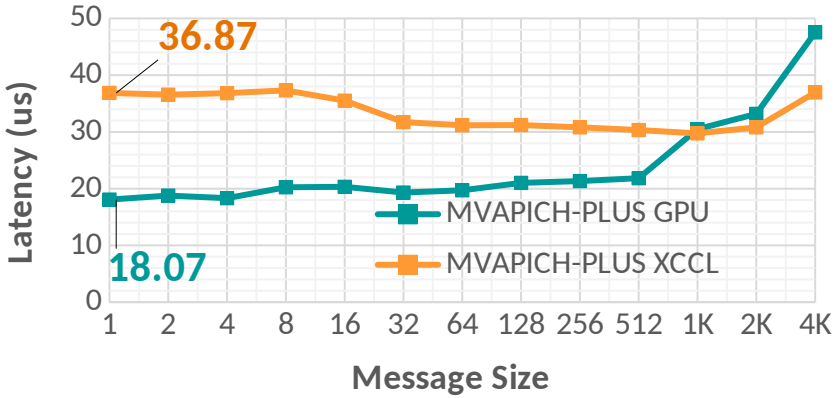
MVAPICH-PLUS GPU Optimized for Collectives – NVIDIA + IB

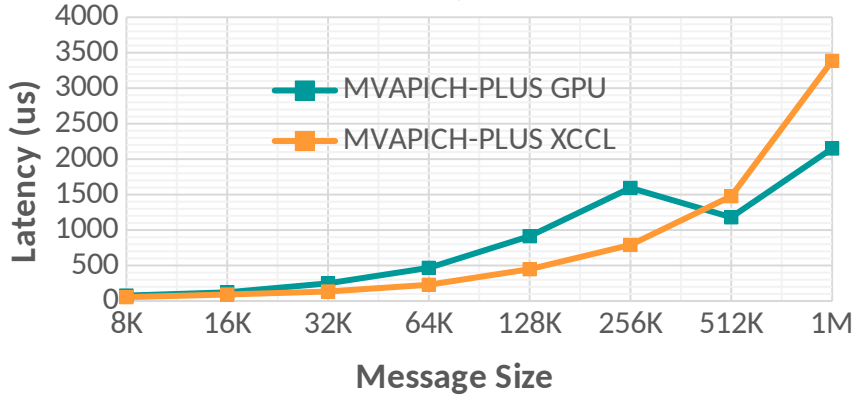
MPI_Gather:



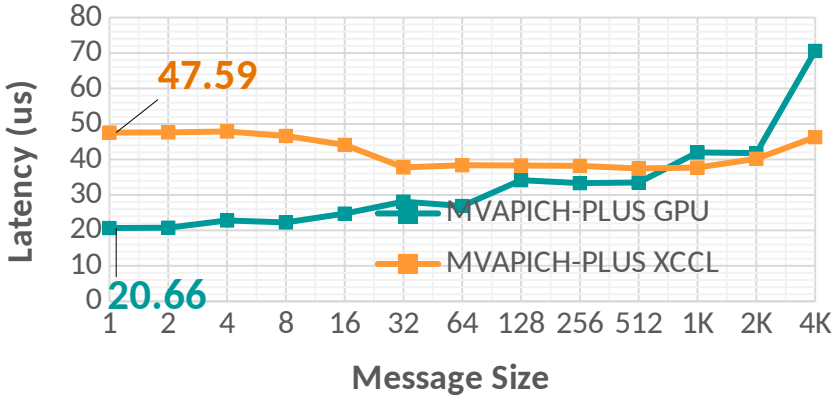


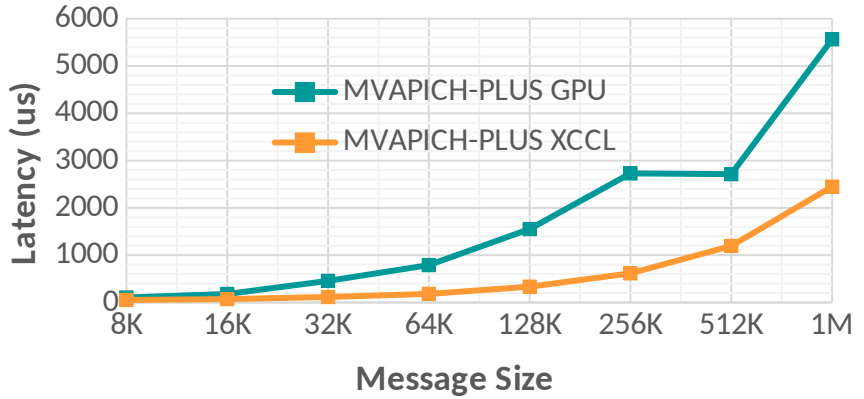
MPI_Scatter:





MPI_Alltoall:

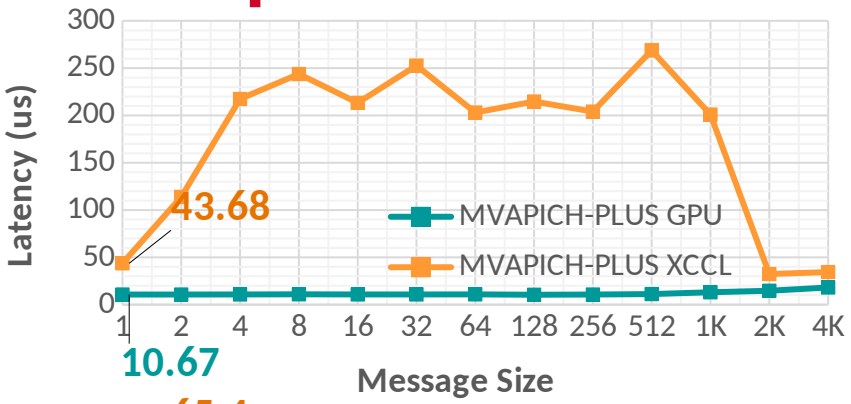


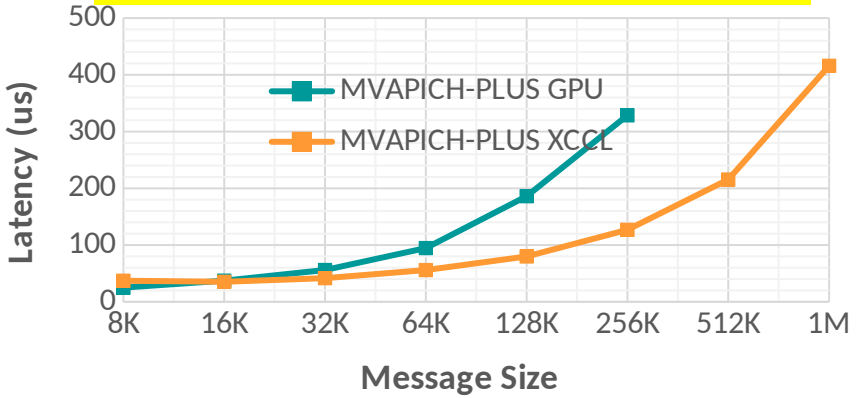


NVIDIA A100 GPUs (8 GPUS – 4 Nodes, 2 GPUs per Node) + CUDA 12.0

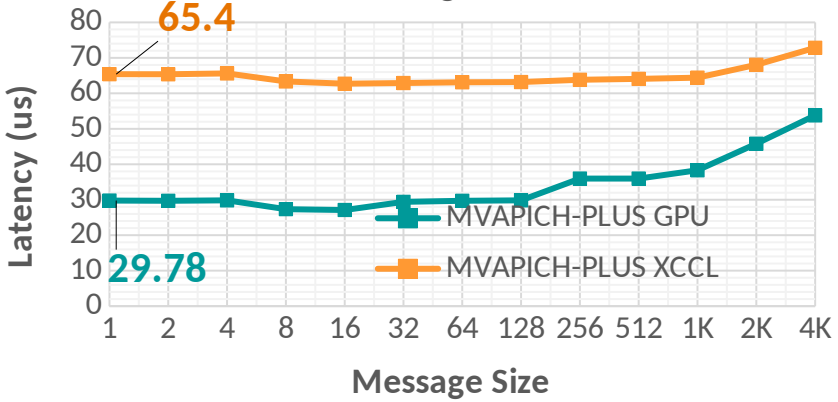
MVAPICH-PLUS GPU Optimized for Collectives – AMD + Slingshot-11

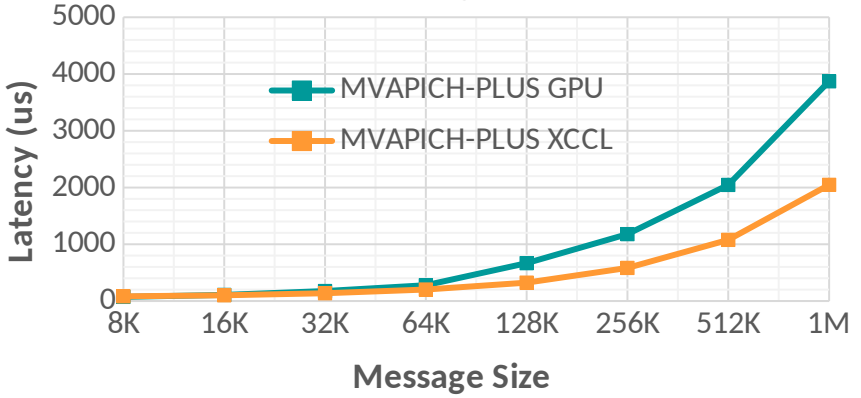
MPI_Gather:



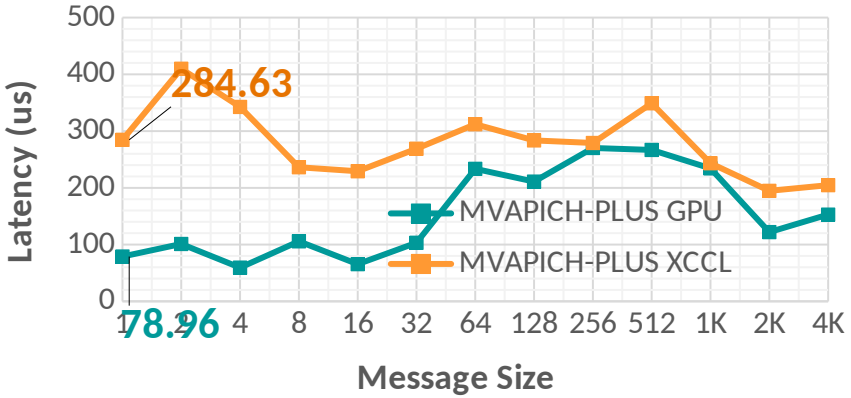


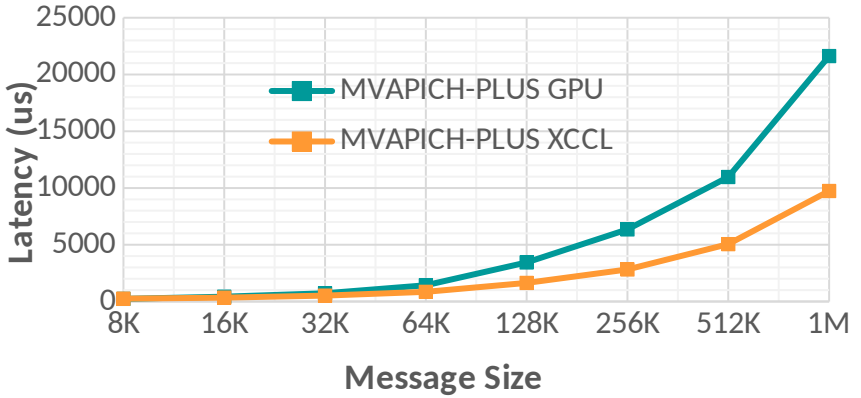
MPI_Scatter:





MPI_Alltoall:





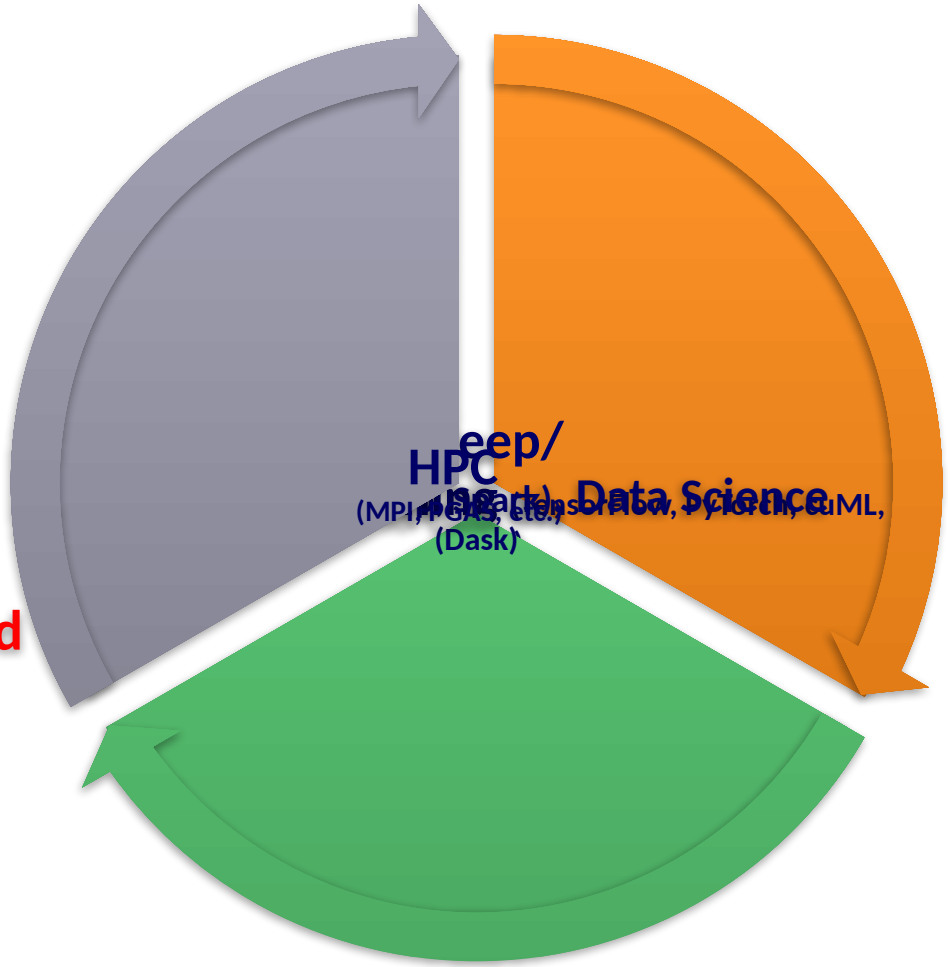
AMD MI250 GPUs (16 GPUS – 2 Nodes, 8 GPUs per Node) + ROCm 5.1.0

Outline

- Brief Overview of the MVAPICH Project
- **Features of Recent Releases**
  - MVAPICH2 2.3.7
  - MVAPICH 3.0b
  - MVAPICH2-J
  - MVAPICH2-X-AWS 2.3.7 and Cloud Deployments
  - MVAPICH2-GDR 2.3.7 for GPUs
  - MVAPICH-Plus 3.0a2
  - **Converged software stack based on MVAPICH**
    - **Support for DL (HiDL), ML (MPI4cuML), Big Data (MPI4Spark), and Data Science (MPI4Dask)**
  - OSU Micro-Benchmarks (OMB)
  - InfiniBand Network Analysis and Monitoring (INAM)
  - Applications: Best Practices
- Upcoming Features
- Conclusions

MPI-Driven Solutions for AI, Big Data and Data Science

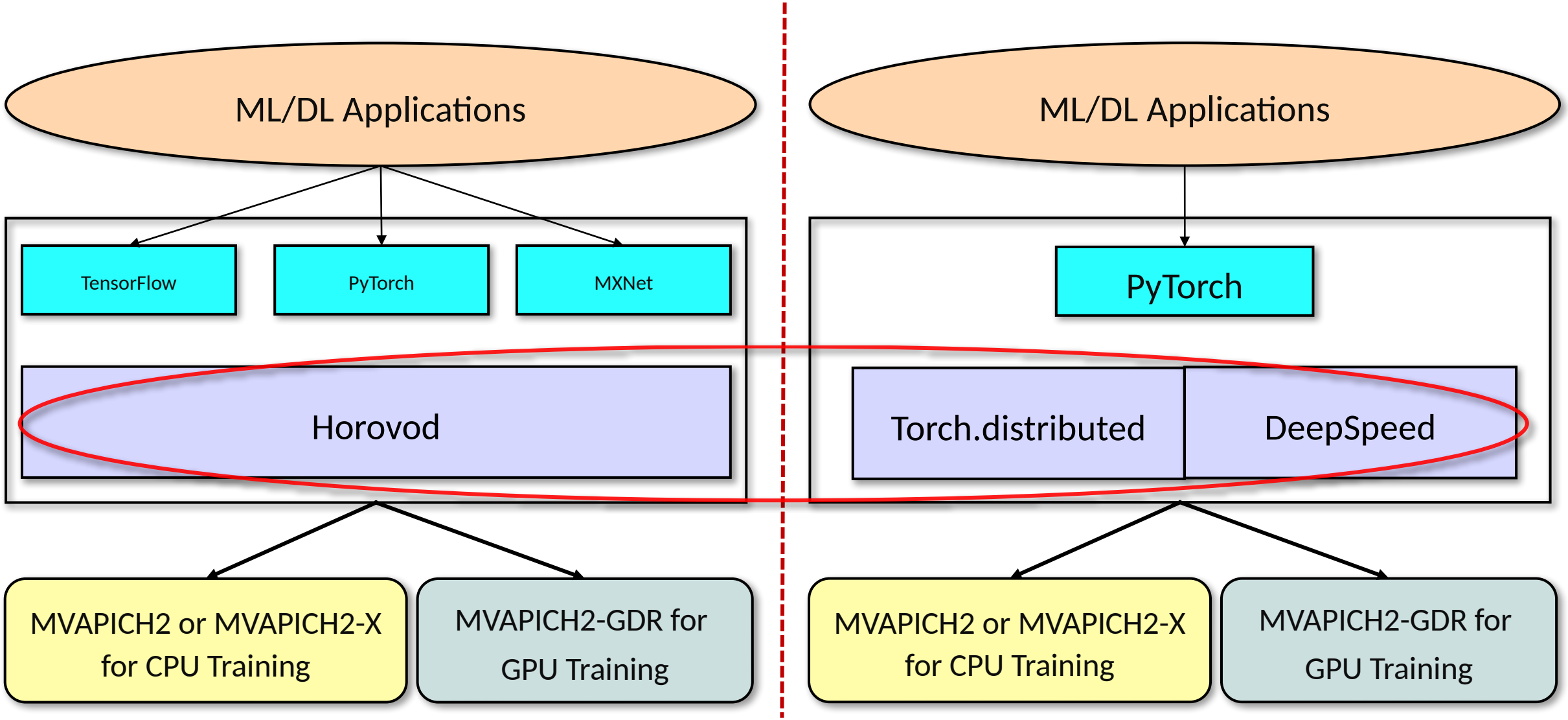
Convergence of HPC, Deep/Machine Learning, and Data Science!



Four Different Modules for Converged Software Stack

- Support for
  - Deep Learning (HiDL)
  - Machine Learning (MPI4cuML)
  - Big Data (MPI4Spark)
  - Data Science (MPI4Dask)

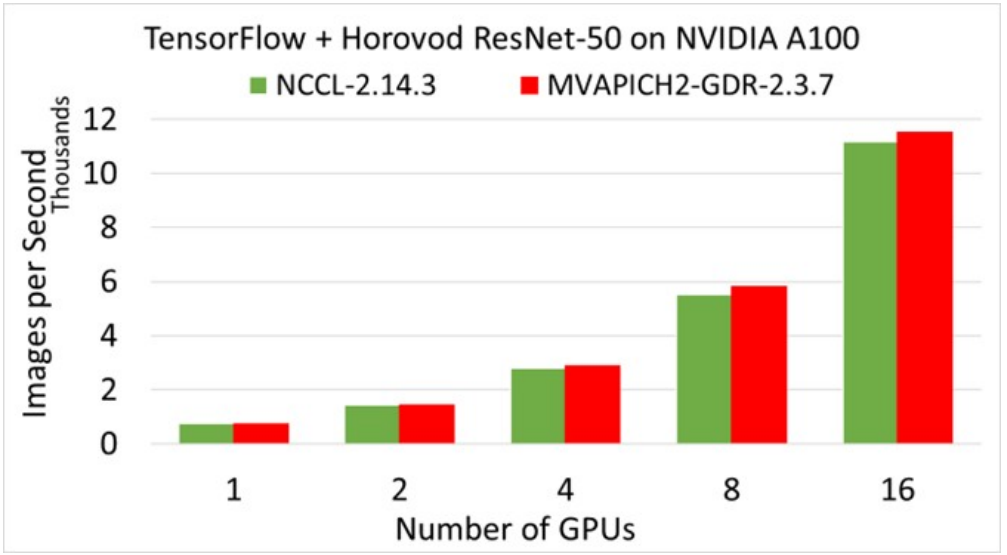
MVAPICH2 (MPI)-driven Infrastructure for ML/DL Training

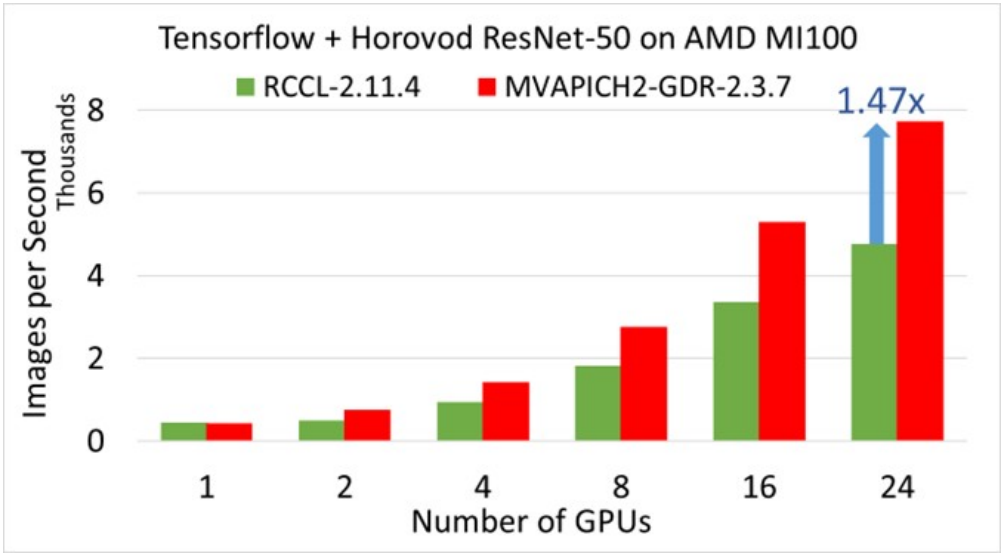


More details available from: http://hidl.cse.ohio-state.edu

HiDL Software Stack Release v1.0

- Based on Horovod
- Optimized support for MPI controller in deep learning workloads
- Efficient large-message collectives (e.g. Allreduce) on various CPUs and GPUs
- GPU-Direct algorithms for collective operations (including those commonly used for data- and model-parallelism, e.g. Allgather and Alltoall)
- Support for fork safety
- Exploits efficient large message collectives
- Compatible with
  - Mellanox InfiniBand adapters (EDR, FDR, HDR)
  - Various x86-based multi-core CPUs (AMD and Intel)
  - NVIDIA A100, V100, P100, Quadro RTX 5000 GPUs
  - CUDA [9.x, 10.x, 11.x] and cuDNN [7.5.x, 7.6.x, 8.0.x, 8.2.x, 8.4.x]
  - AMD MI100 GPUs
  - ROCm [5.1.x]

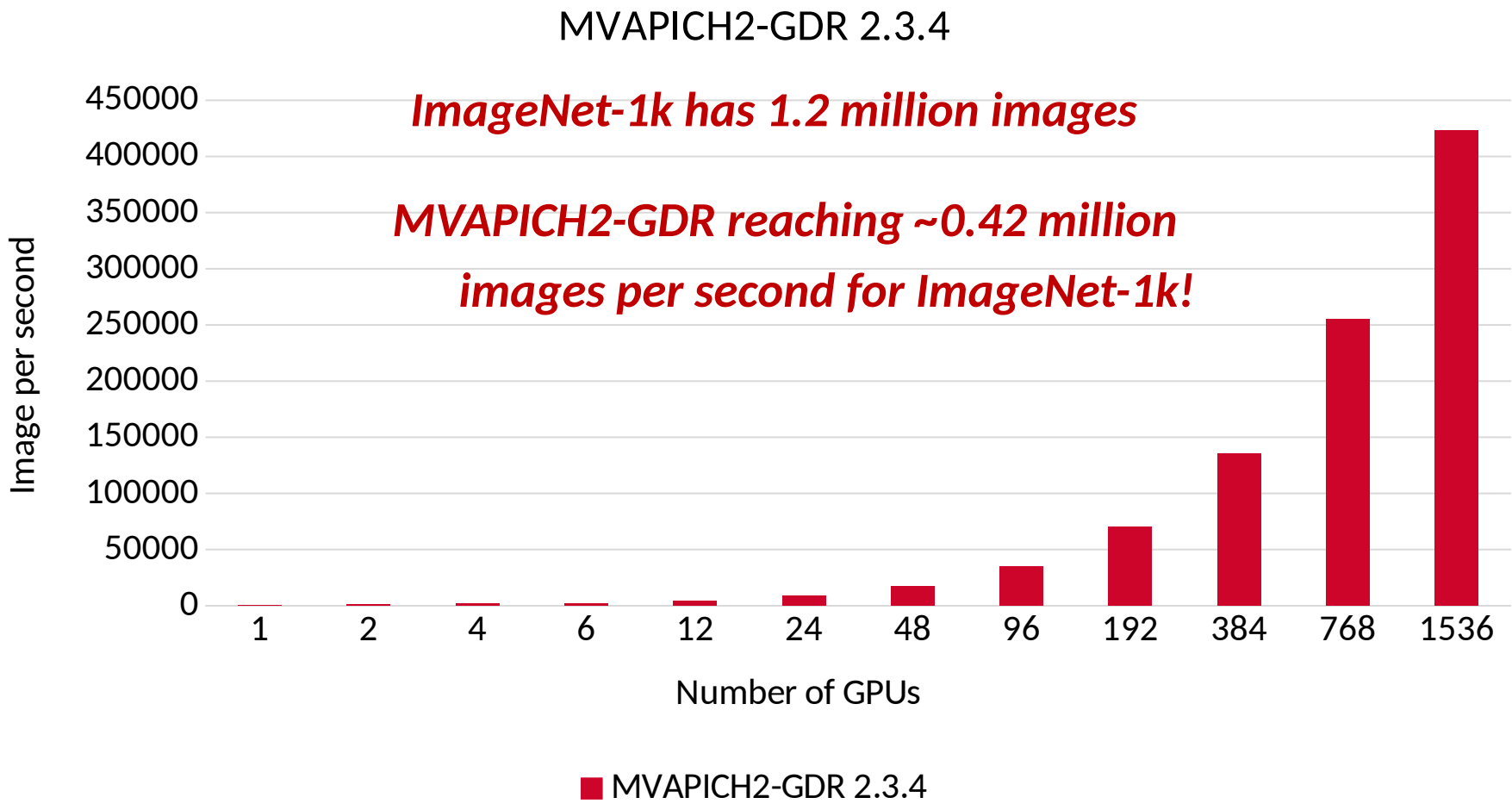




For more details: http://hidl.cse.ohio-state.edu/userguide/horovod/

Distributed TensorFlow on ORNL Summit (1,536 GPUs)

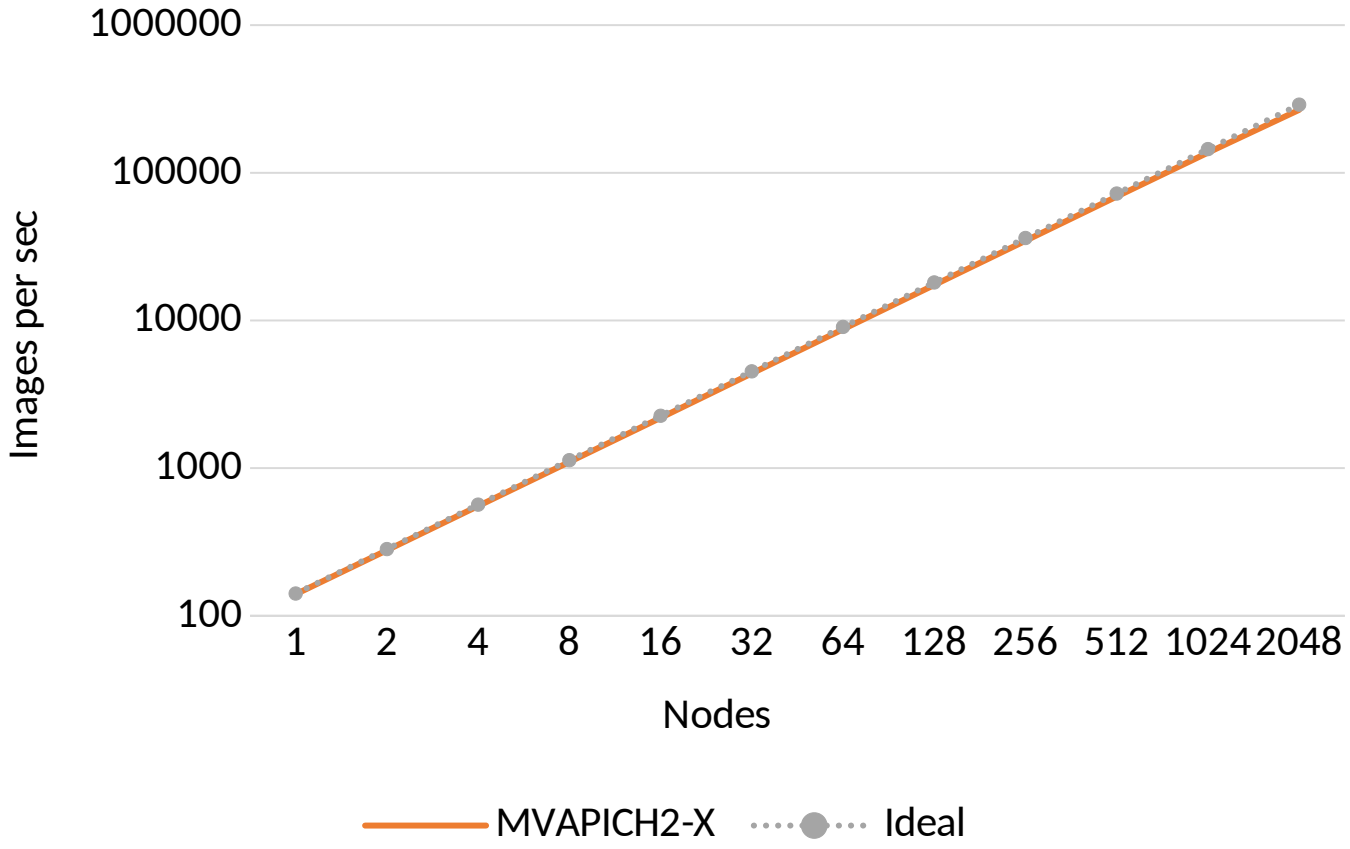
- ResNet-50 Training using TensorFlow benchmark on SUMMIT -- 1536 Volta GPUs!
- 1,281,167 (1.2 mil.) images
- Time/epoch = 3 seconds
- Total Time (90 epochs) = 3 x 90 = 270 seconds = **4.5 minutes!**



Platform: The Summit Supercomputer (#2 on Top500.org) – 6 NVIDIA Volta GPUs per node connected with NVLink, CUDA 10.1

Distributed TensorFlow on TACC Frontera (2048 CPU nodes)

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



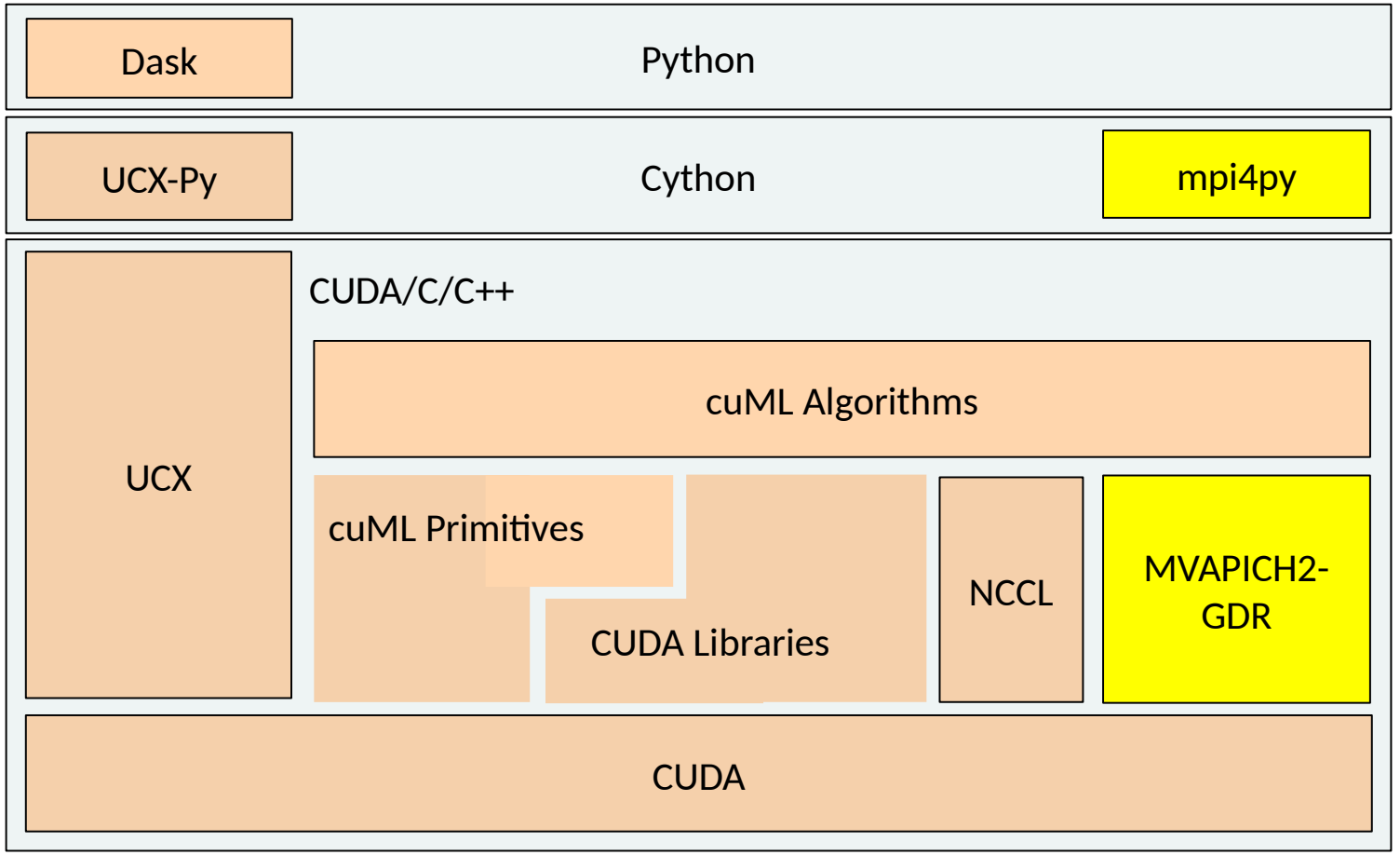
A. Jain, A. A. Awan, H. Subramoni, DK Panda, "Scaling TensorFlow, PyTorch, and MXNet using MVAPICH2 for High-Performance Deep Learning on Frontera", DLS '19 (SC '19 Workshop).

Accelerating cuML (Machine Learning) with MVAPICH2-GDR

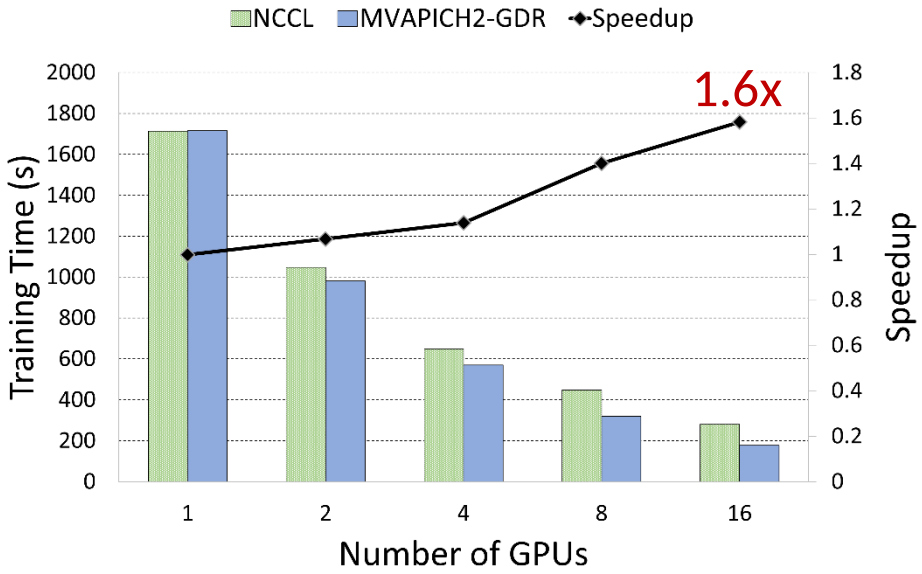
- Utilize MVAPICH2-GDR (with mpi4py) as communication backend during the training phase (the fit() function) in the Multi-node Multi-GPU (MNMG) setting over cluster of GPUs
- Communication primitives:
  - Allreduce
  - Reduce
  - Broadcast
- Exploit optimized collectives

MPI4cuML 0.5 release

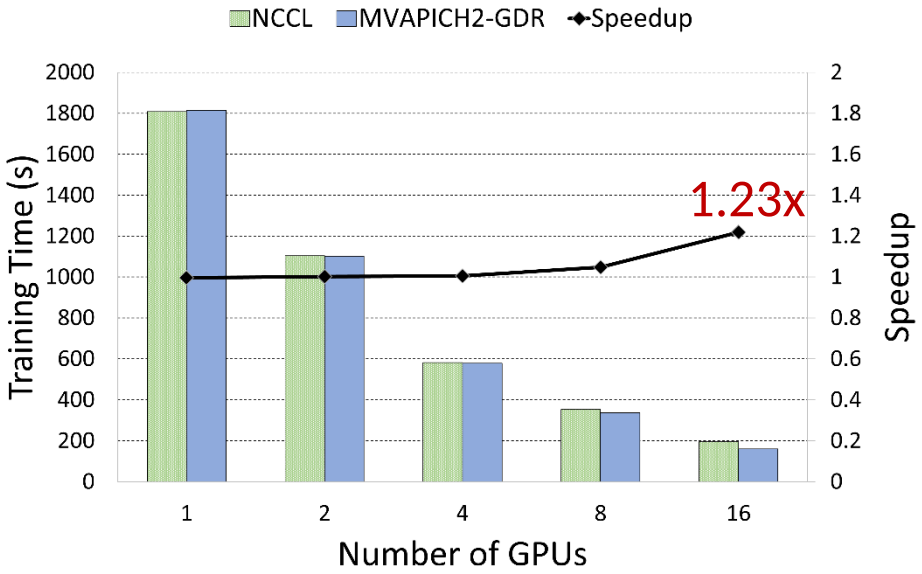
(http://hidl.cse.ohio-state.edu)



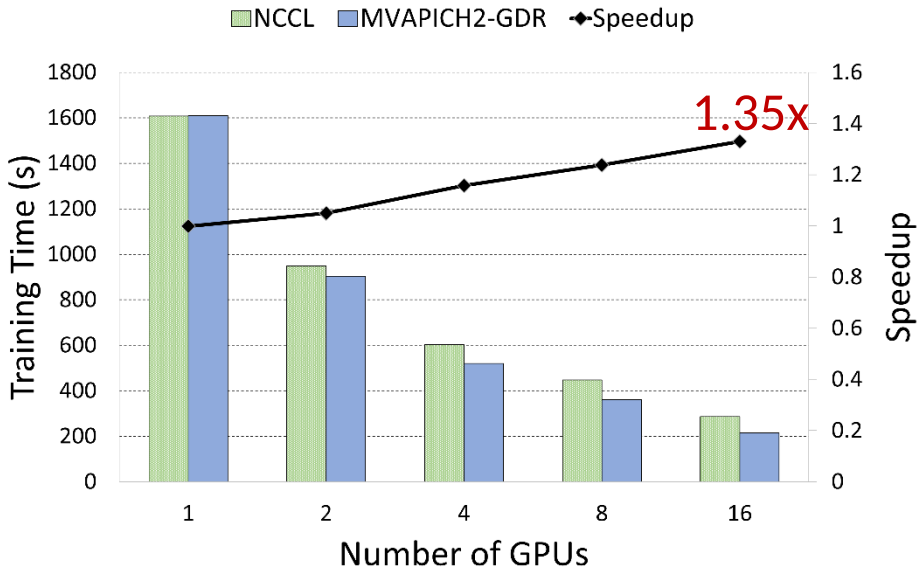
K-Means



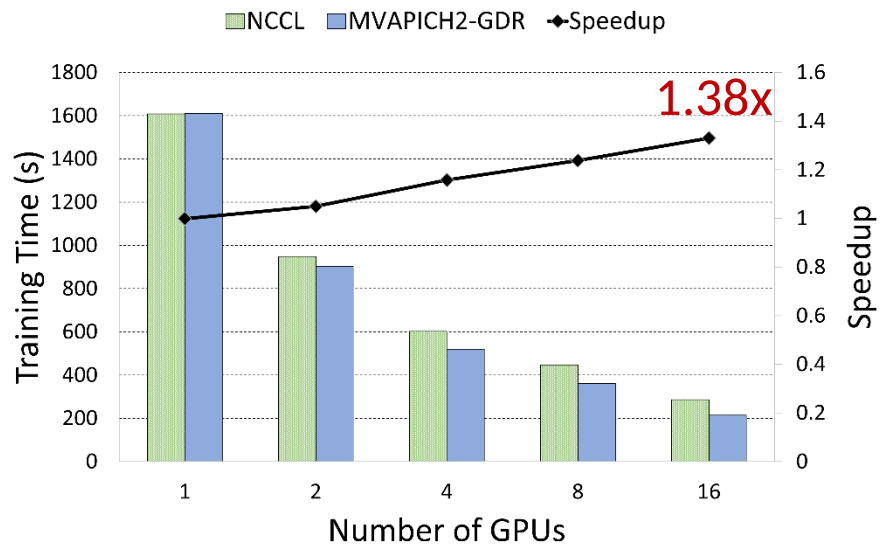
Linear Regression



Nearest Neighbors



Truncated SVD



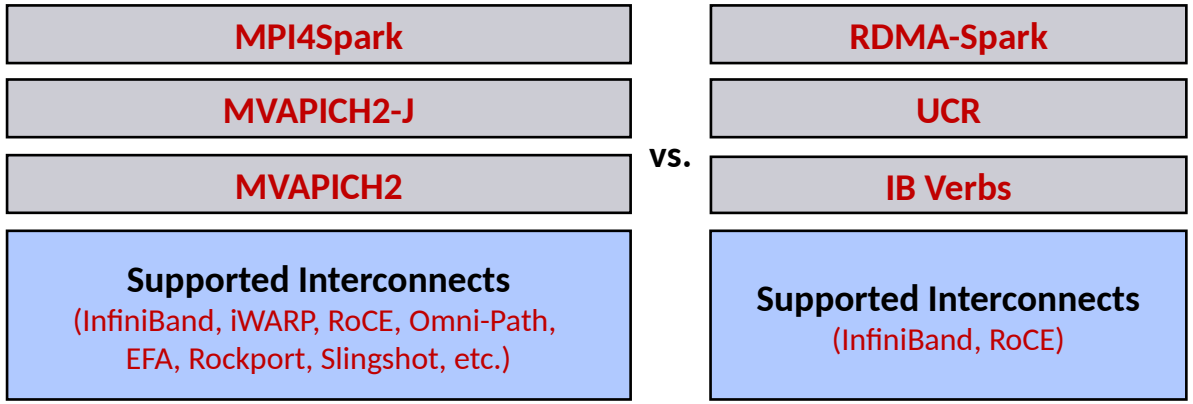
MPI4cuML 0.5

Expanse GPU System	
CPU Model	Intel Xeon Gold 6248
CPU Core Info	2x20 @ 2.5Ghz
Memory	384 GB
Interconnect	Infiniband HDR (200 Gbps)
OS	Rocky Linux 8.5
GPU	NVIDIA V100 (4/Node)
CUDA	CUDA 11.2

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

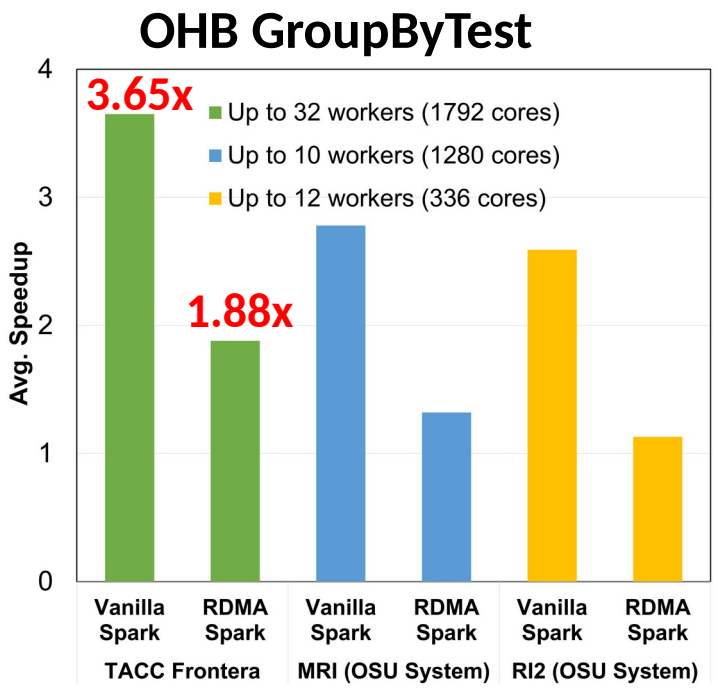
MPI4Spark: Supporting Spark over MVAPICH2

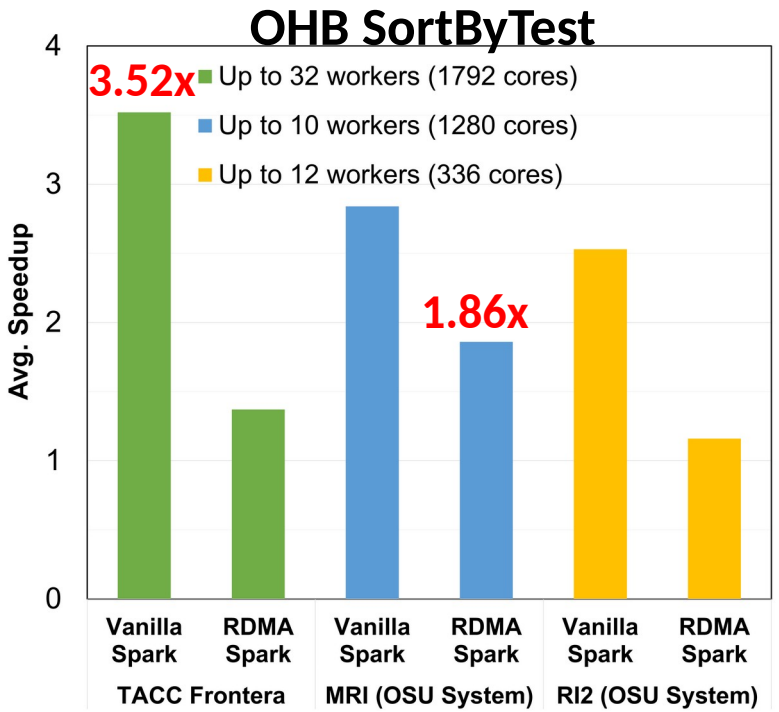
- The current approach is different from its predecessor design, RDMA-Spark (http://hibd.cse.ohio-state.edu)
  - RDMA-Spark supports only InfiniBand and RoCE
  - Requires new designs for new interconnect
- MPI4Spark supports multiple interconnects/systems through a common MPI library
  - Such as InfiniBand (IB), Intel Omni-Path (OPA), HPE Slingshot, RoCE, and others
  - No need to re-design the stack for a new interconnect as long as the MPI library supports it



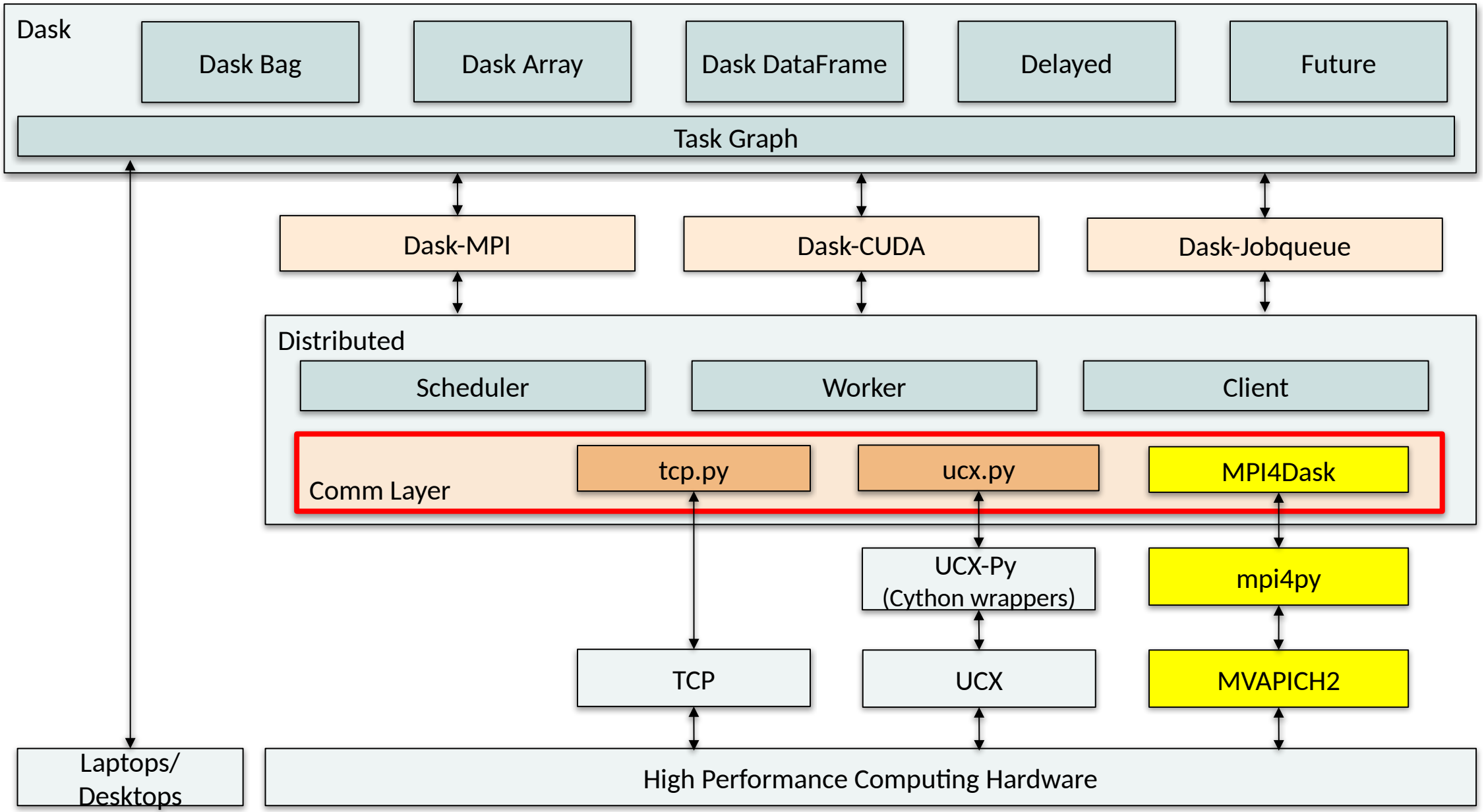
MPI4Spark: Relative Speedups to Vanilla Spark and RDMA-Spark

System Name	Nodes Used	Processor	Cores Used	Sockets	Cores/socket	RAM	Interconnect
TACC Frontera	34	Xeon Platinum	1792	2	28	192 GB	HDR (100G)
RI2 (OSU System)	14	Xeon Broadwell	336	2	14	128 GB	EDR (100G)
MRI (OSU System)	12	AMD EPYC 7713	1280	2	64	264 GB	200 Gb/sec (4X HDR)
TACC Stamepede2	10	Xeon Platinum	384	2	28	192 GB	Intel Omni-Path (100G)





Accelerating Dask with MPI4Dask



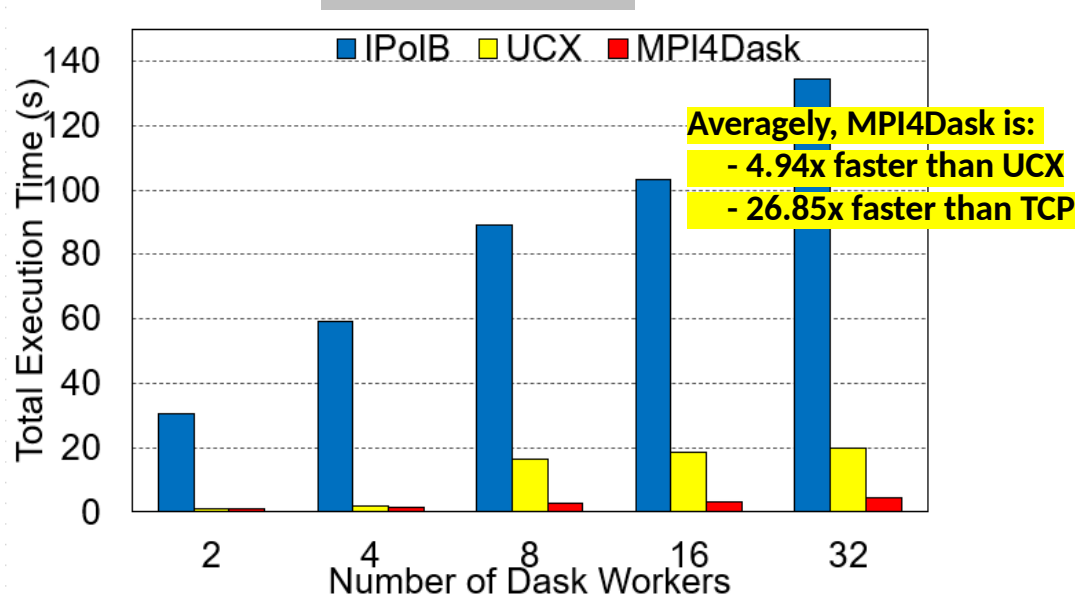
cuDF Merge Benchmark on the Cambridge Wilkes-3 System

- GPU-based Operation: , using persist
  - Merge two GPU data frames, each with length of 32*1e8
  - Compute() will gather the data from all worker nodes to the client node, and make a copy on the host memory.
  - Persist() will leave the data on its current nodes without any gathering

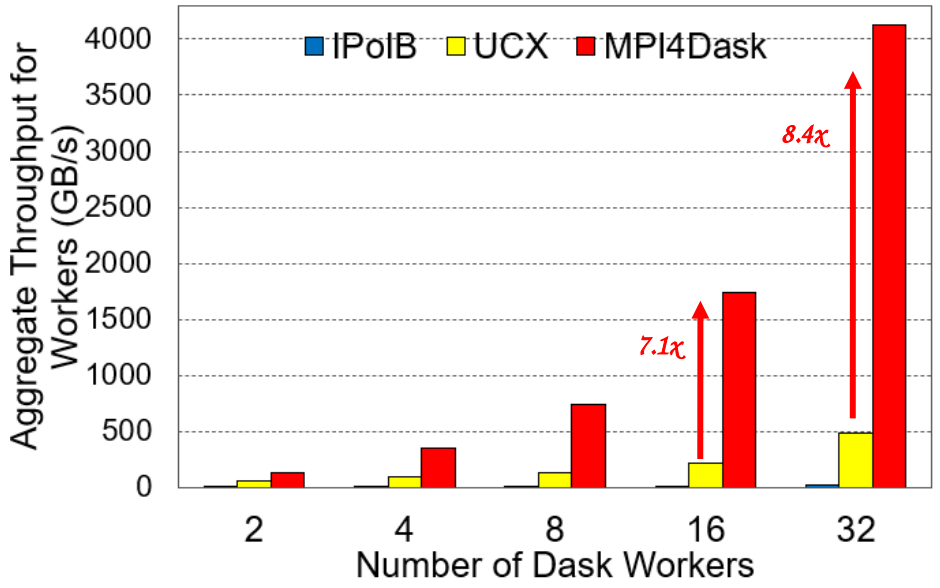
Wilke3 GPU System:

- **80 nodes**
- **2x AMD EPYC 7763 64-core Processors**
- **1000 GiB RAM**
- **Dual-rail Mellanox HDR200 IB**
- **4x NVIDIA A100 SXM4 80 GB**

Execution Time



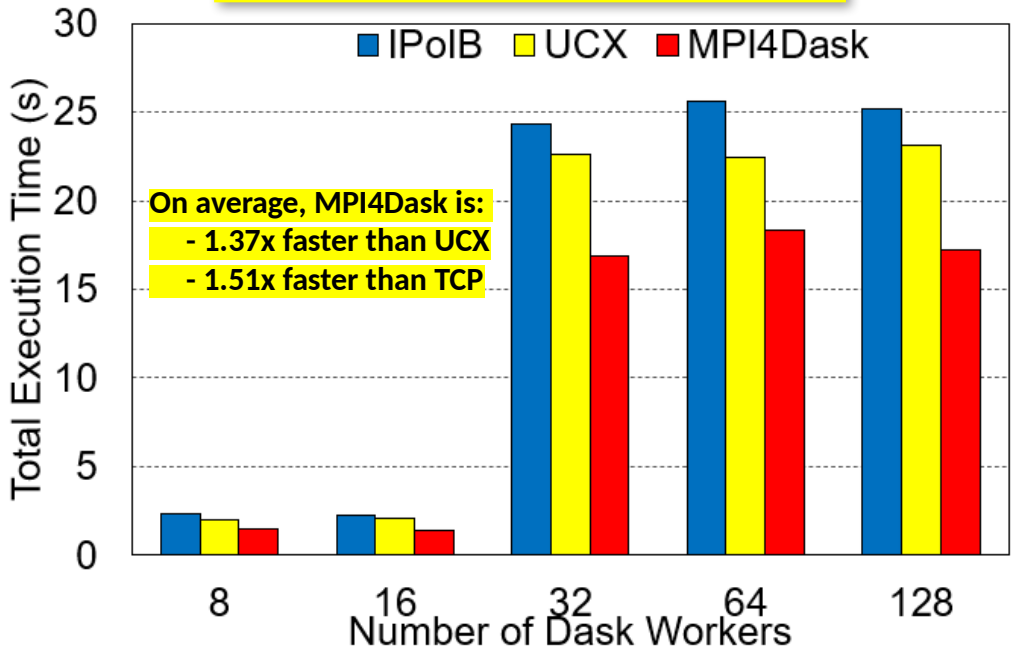
Aggregated Throughput



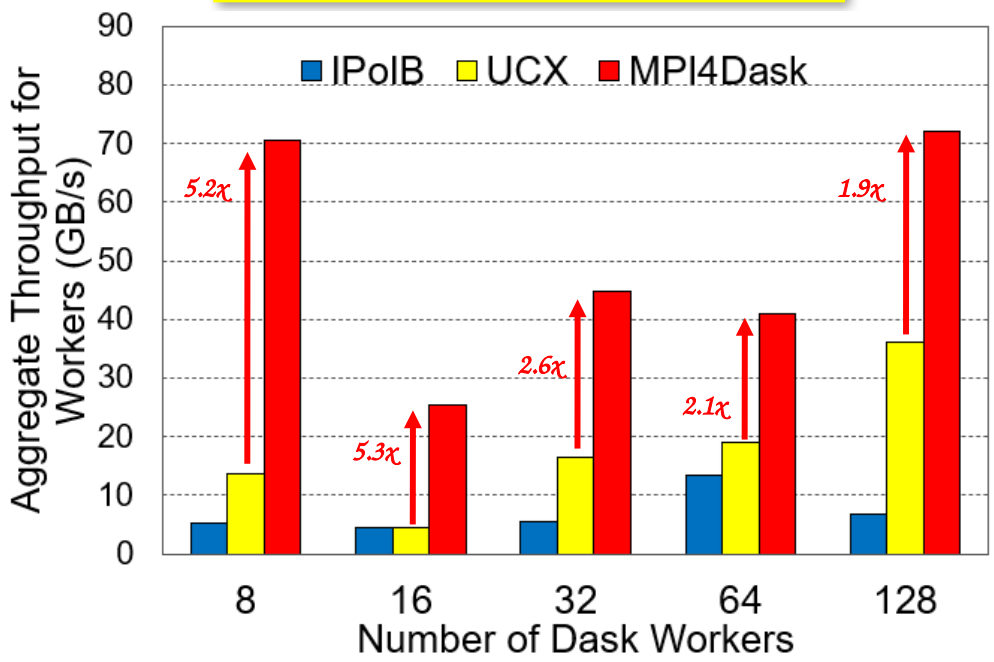
MPI4Dask 0.3, Dask 2022.8.1, Distributed, 2022.8.1, MVAPICH2-GDR 2.3.7, UCX v1.13.1, UCX-py 0.27.00

NumPy Array Slicing Benchmark on TACC Frontera CPU System

1.26x better on average



3.17x better on average



From 32 workers, we increase array size by 16 times

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 0.3 release

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

Outline

- Brief Overview of the MVAPICH Project
- **Features of Recent Releases**
  - MVAPICH2 2.3.7
  - MVAPICH 3.0b
  - MVAPICH2-J
  - MVAPICH2-X-AWS 2.3.7 and Cloud Deployments
  - MVAPICH2-GDR 2.3.7 for GPUs
  - MVAPICH-Plus 3.0a2
  - Converged software stack based on MVAPICH
    - Support for DL (HiDL), ML (MPI4cuML), Big Data (MPI4Spark), and Data Science (MPI4Dask)
  - **OSU Micro-Benchmarks (OMB)**
  - InfiniBand Network Analysis and Monitoring (INAM)
  - Applications: Best Practices
- Upcoming Features
- Conclusions

OSU Microbenchmarks

- Available since 2004
- Suite of microbenchmarks to study communication performance of various programming models
- 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 accelerator-based architectures
  - Compute Unified Device Architecture (CUDA)
  - OpenACC Application Program Interface
- Part of various national resource procurement suites like NERSC-8 / Trinity Benchmarks
- Continuing to add support for newer primitives and features (since OMB 6.0)
  - Java Binding Support
  - Python Support
- Please visit the following link for more information: http://mvapich.cse.ohio-state.edu/benchmarks/

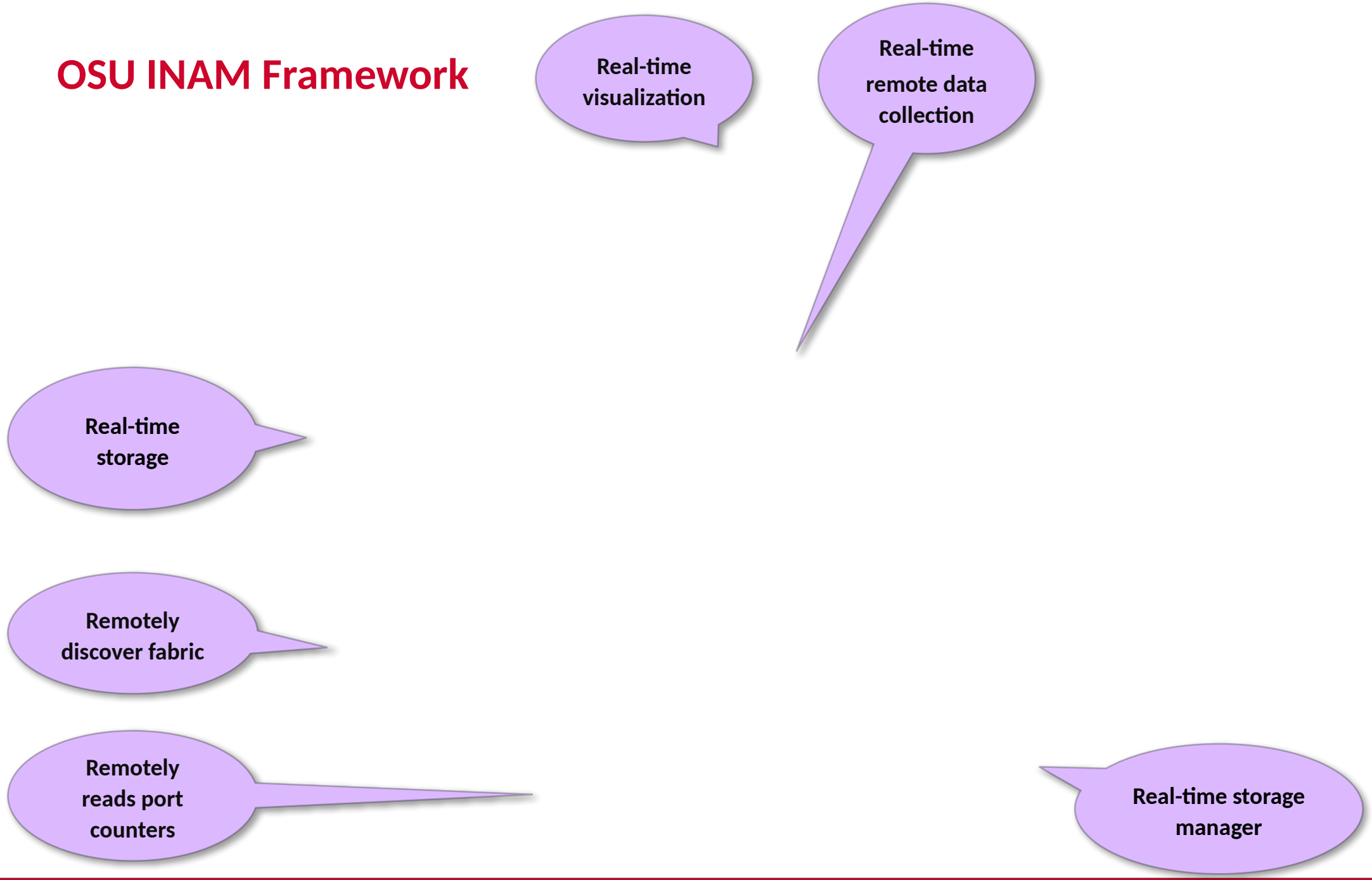
OSU Micro Benchmarks v7.2

- New features since MUG'22
  - Add MPI-4 session based initialization support to the following benchmarks.
    - Pt2pt, Collective, One-sided, Neighborhood
  - Add MPI_IN_PLACE support for following blocking and non-blocking collectives.
  - Add an option to set root rank for rooted blocking and non-blocking collectives.
  - Add new blocking and non-blocking neighborhood collective benchmarks with support for CPU buffers.
  - Add new benchmarks for point-to-point persistent communication primitives.
  - Add PAPI support for the following MPI benchmarks.
    - Pt2pt, Collective, One-sided
  - Add support for benchmarking non-blocking Reduce-Scatter.
  - Add graphing support to the following MPI benchmarks.
    - Pt2pt, Collective, One-sided
  - Add support for benchmarking blocking Alltoallw.
  - Add support for derived data types in MPI pt2pt benchmarks.
  - Extend collective and point-to-point benchmarks to support different MPI datatypes.
  - Add support for derived data types in MPI blocking and non-blocking collective benchmarks

Outline

- Brief Overview of the MVAPICH Project
- **Features of Recent Releases**
  - MVAPICH2 2.3.7
  - MVAPICH 3.0b
  - MVAPICH2-J
  - MVAPICH2-X-AWS 2.3.7 and Cloud Deployments
  - MVAPICH2-GDR 2.3.7 for GPUs
  - MVAPICH-Plus 3.0a2
  - Converged software stack based on MVAPICH
    - Support for DL (HiDL), ML (MPI4cuML), Big Data (MPI4Spark), and Data Science (MPI4Dask)
  - OSU Micro-Benchmarks (OMB)
  - **InfiniBand Network Analysis and Monitoring (INAM)**
  - Applications: Best Practices
- Upcoming Features
- Conclusions

OSU INAM Framework



Overview of OSU InfiniBand Network Analysis and Monitoring (INAM) Tool

- **A network monitoring and analysis tool that is capable of analyzing traffic on the InfiniBand network with inputs from the MPI runtime**
  - http://mvapich.cse.ohio-state.edu/tools/osu-inam/
- **Monitors IB clusters in real time by querying various subnet management entities and gathering input from the MPI runtimes**
- **Capability to analyze and profile node-level, job-level and process-level activities for MPI communication**
  - Point-to-Point, Collectives and RMA
- **Ability to filter data based on type of counters using "drop down" list**
- **Remotely monitor various metrics of MPI processes at user specified granularity**
- **"Job Page" to display jobs in ascending/descending order of various performance metrics in conjunction with MVAPICH2-X**
- **Visualize the data transfer happening in a "live" or "historical" fashion for entire network, job or set of nodes**
- **Sub-second port query and fabric discovery in less than 10 mins for ~2,000 nodes**

- **OSU INAM 1.0 released (11/10/2022)**
  - Enhanced the UI by making asynchronous API calls for data loading
    - Significantly improved page load performance
  - Support for continuous queries to improve visualization performance
  - Support for MySQL and InfluxDB as database backends
    - Enhanced database insertion using InfluxDB
  - Support for PBS and SLURM job scheduler as config time
    - Support for SLURM multi-cluster configuration
  - Enable emulation mode to allow users to test OSU INAM tool in a sandbox environment without actual deployment
  - Generate email notifications to alert users when user defined events occur
  - Support to display node-/job-level CPU, Virtual Memory, and Communication Buffer utilization information for historical jobs
  - Support to handle multiple job schedulers on the same fabric
  - Support to collect and visualize MPI_T based performance data
  - Support for MOFED 4.5, 4.9+, 5+
  - Support for adding user-defined labels for switches to allow better readability and usability
  - Support authentication for accessing the OSU INAM webpage
  - Optimized webpage rendering and database fetch/purge capabilities

Outline

- Brief Overview of the MVAPICH Project
- **Features of Recent Releases**
  - MVAPICH2 2.3.7
  - MVAPICH 3.0b
  - MVAPICH2-J
  - MVAPICH2-X-AWS 2.3.7 and Cloud Deployments
  - MVAPICH2-GDR 2.3.7 for GPUs
  - MVAPICH-Plus 3.0a2
  - Converged software stack based on MVAPICH
    - Support for DL (HiDL), ML (MPI4cuML), Big Data (MPI4Spark), and Data Science (MPI4Dask)
  - OSU Micro-Benchmarks (OMB)
  - InfiniBand Network Analysis and Monitoring (INAM)
  - **Applications: Best Practices**
- Upcoming Features
- Conclusions

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.

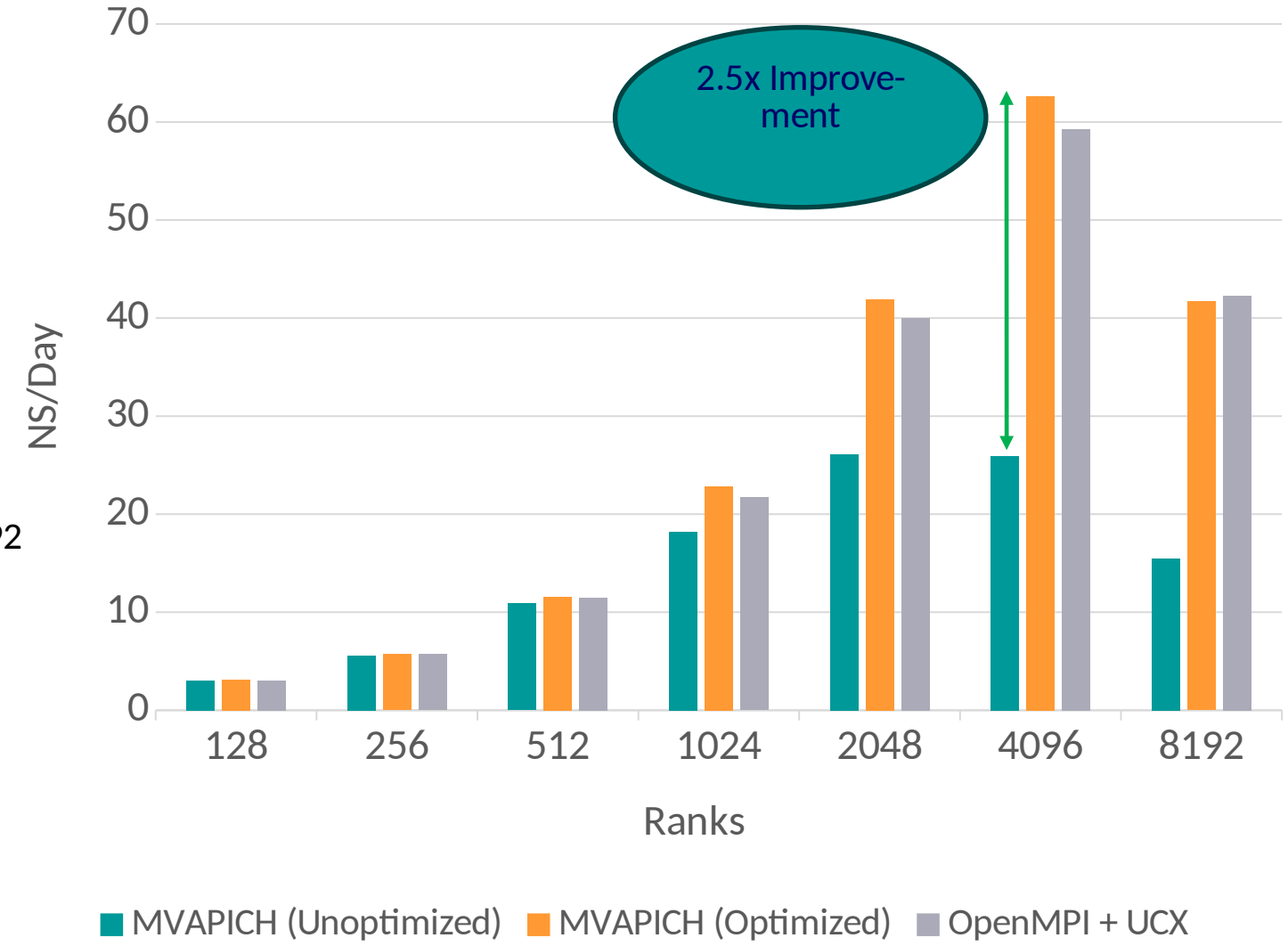
Case Study: GROMACS – Performance Engineering with TAU

Diagnosis and workaround found

- Investigate UD communication (read progress poll)
- Use RC to get the desired lead in performance
- Gains:
  - 2.5x improvement over MVAPICH baseline
  - 15% compared to OpenMPI default RC
- Update the following parameter for GROMACS runs

`MV2_HYBRID_ENABLE_THRESHOLD = 8192`

this will enable UD-hybrid communication after the 8192 threshold*



*For more details check user-guide:
https://mvapich.cse.ohio-state.edu/static/media/mvapich/mvapich2-userguide.html#:~:text=use%20any%20HugePages.-,11.110,-MV2_HYBRID_ENABLE_THRESHOLD

Outline

- Brief Overview of the MVAPICH Project
- Features of Recent Releases
- **Upcoming Features**
  - **MVAPICH and MVAPICH-Plus 3.0 series**
  - MVAPICH-Plus for AI-Accelerator Chips (Habana/Gaudi)
  - MVAPICH and OMB for FPGA
  - INAM
  - MPI4DL for Deep Learning
  - MPI4Spark
  - Conversational AI Interface (SAI)
- Conclusions

Plans for MVAPICH 3.0 and MVAPICH-Plus 3.0

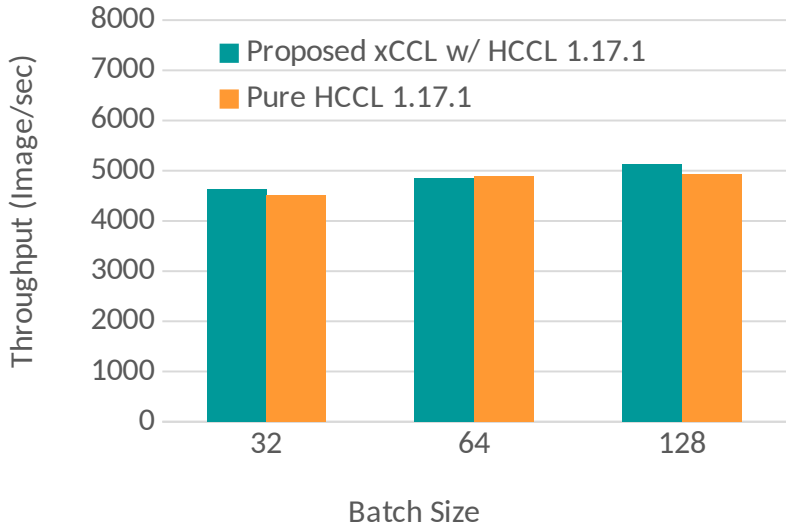
- MVAPICH 3.0
  - Superset features compared to MVAPICH2 2.3.7 for all interconnects
  - Users can move to MVAPICH 3.0 for production runs
  - New MPI 4.0 features on CPU-CPU communication

- MVAPICH-Plus 3.0
  - Superset features compared to (MVAPICH2-GDR 2.3.7 + MVAPICH2-X) for all interconnects and all GPUs (NVIDIA and AMD)
  - Users can move to MVAPICH-Plus 3.0 for production runs
  - Support for Intel GPUs
  - New MPI 4.0 features on GPU-GPU communication

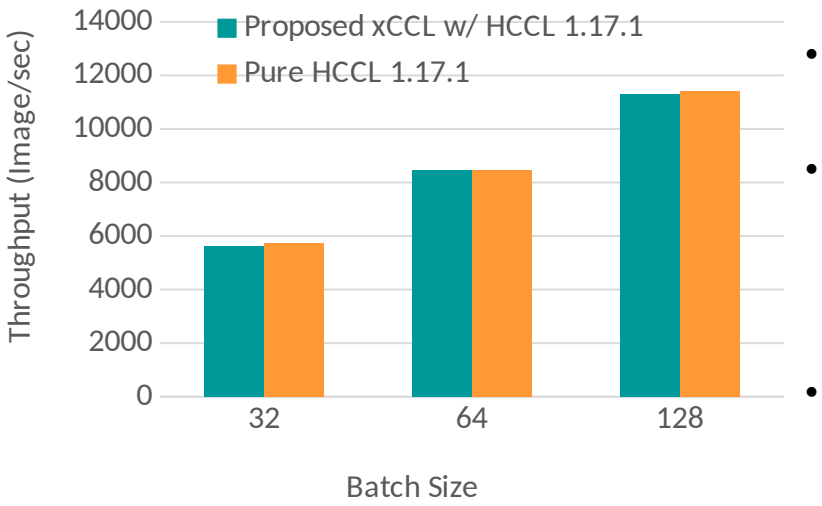
Outline

- Brief Overview of the MVAPICH Project
- Features of Recent Releases
- **Upcoming Features**
  - MVAPICH and MVAPICH-Plus 3.0 series
  - **MVAPICH-Plus for AI-Accelerator Chips (Habana/Gaudi)**
  - MVAPICH and OMB for FPGA
  - INAM
  - MPI4DL for Deep Learning
  - MPI4Spark
  - Conversational AI Interface (SAI)
- Conclusions

MVAPICH-xCCL Design on Habana: Application-Level Evaluation



1 Node, 8 HPUs



4 Nodes, 32 HPUs

- The container image has already contained prebuilt Habaha TensorFlow and Horovod.
- The computing kernel is replaced by Habana ops through TensorFlow custom ops, the com munication layer in Horovod is implemented by HCCL directly.
- Modify the Horovod communication by replacing all `hcclAllreduce` calls with `MPI_Allreduce` operations.

- On single node, xCCL provides 5139 img/sec throughput with batch size 128, and it is close to the throughput of 4936 img/sec using pure HCCL (4% overhead).
- On multiple nodes (4 nodes), both xCCL and pure HCCL reach the throughput of 11300 img/sec where the overhead is less than 1%
- This evaluation proves that our xCCL designs can be easily extended to new architectures and collective communication libraries with negligible overheads.

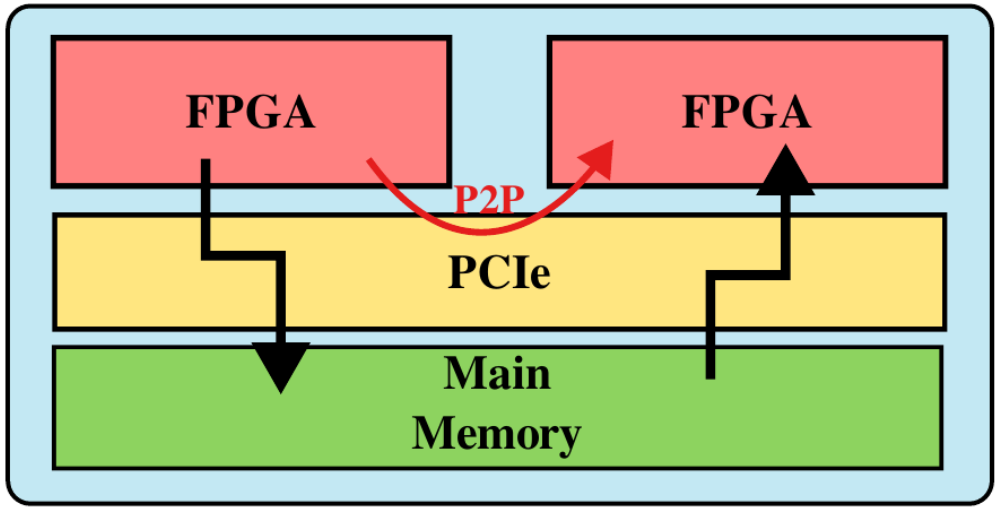
- More details
  - Chen-Chun's Talk (yesterday)
  - Mahidhar's Talk (tomorrow)

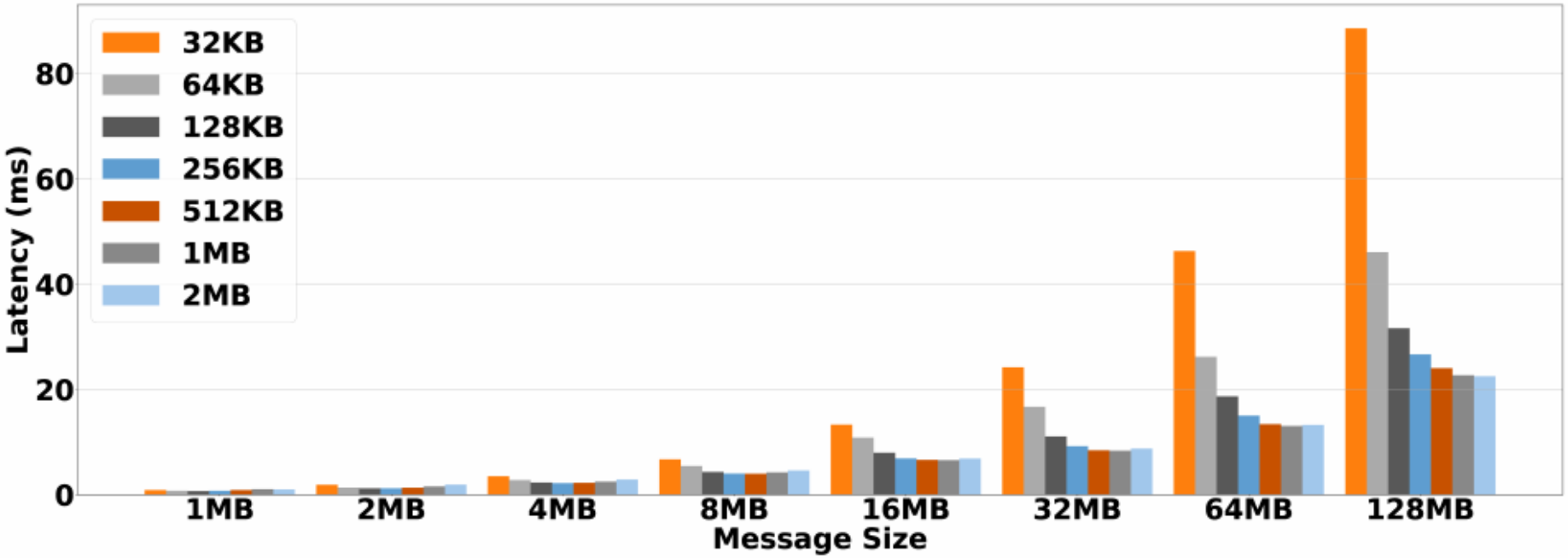
Outline

- Brief Overview of the MVAPICH Project
- Features of Recent Releases
- **Upcoming Features**
  - MVAPICH and MVAPICH-Plus 3.0 series
  - MVAPICH-Plus for AI-Accelerator Chips (Habana/Gaudi)
  - **MVAPICH and OMB for FPGA**
  - INAM
  - MPI4DL for Deep Learning
  - MPI4Spark
  - Conversational AI Interface (SAI)
- Conclusions

Optimized MVAPICH-FPGA Design

- P2P transfers enable data to travel between devices over PCIe without entering host memory
- Extension of OMB for FPGA-based systems





More details in the Short Talk, presented by Nick Contini (Yesterday 4:00-5:30 pm)

Outline

- Brief Overview of the MVAPICH Project
- Features of Recent Releases
- **Upcoming Features**
  - MVAPICH and MVAPICH-Plus 3.0 series
  - MVAPICH-Plus for AI-Accelerator Chips (Habana/Gaudi)
  - MVAPICH and OMB for FPGA
  - **INAM**
  - MPI4DL for Deep Learning
  - MPI4Spark
  - Conversational AI Interface (SAI)
- Conclusions

Plans for INAM

- Support for ClickHouse Database
  - Significantly improve system-wide data collection to sub-second for up to 20,000 nodes
  - Support for up to 32 users to query data in sub-second latency
  - Please refer to Pouya Kousha's poster
- Add support for monitoring ethernet networks by adding support for SNMP
- Release support for visualizing intra-node communication and topology
- Deploy INAM on multiple HPC systems

Outline

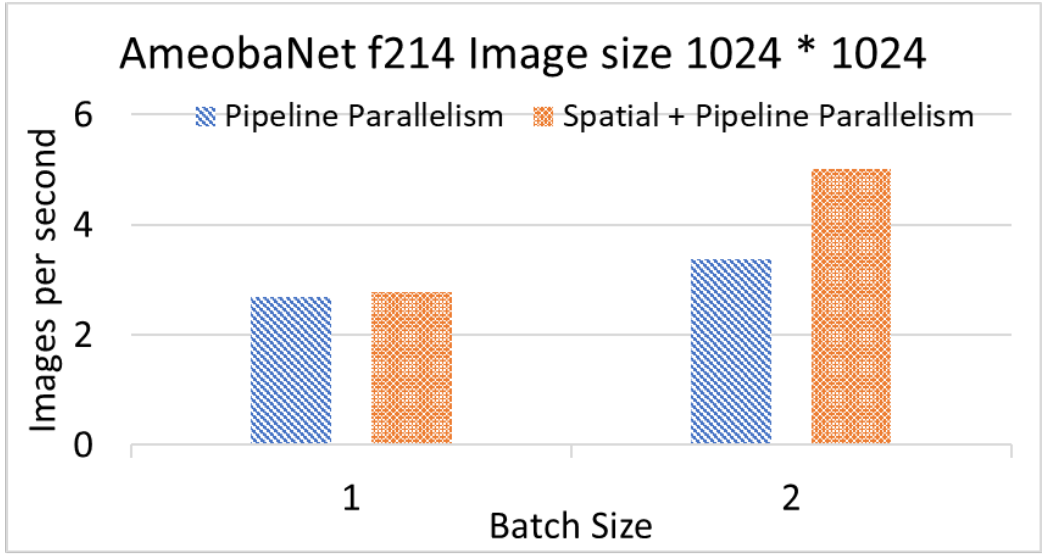
- Brief Overview of the MVAPICH Project
- Features of Recent Releases
- **Upcoming Features**
  - MVAPICH and MVAPICH-Plus 3.0 series
  - MVAPICH-Plus for AI-Accelerator Chips (Habana/Gaudi)
  - MVAPICH and OMB for FPGA
  - INAM
  - **MPI4DL for Deep Learning**
  - MPI4Spark
  - Conversational AI Interface (SAI)
- Conclusions

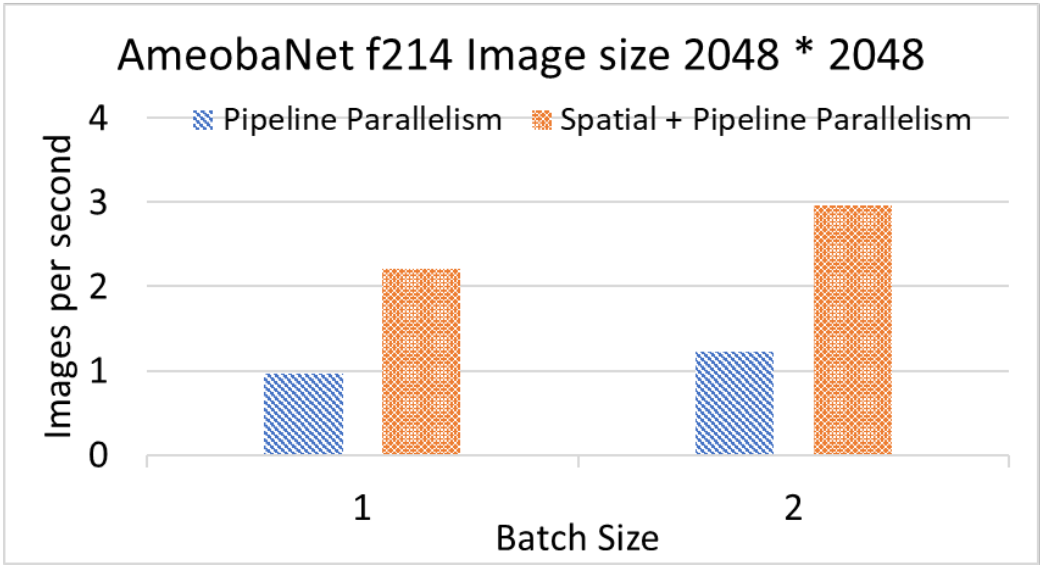
Upcoming Release: MPI4DL v0.5

MPI4DL v0.5 is a distributed and accelerated training framework for very high-resolution images that integrates Spatial Parallelism, Layer Parallelism, and Pipeline Parallelism.

Features:

- Based on PyTorch
- Support for training very high-resolution images
- Distributed training support for:
  - Layer Parallelism (LP)
  - Pipeline Parallelism (PP)
  - Spatial Parallelism (SP)
  - Spatial and Layer Parallelism (SP+LP)
  - Spatial and Pipeline Parallelism (SP+PP)
- Support for different image sizes and custom datasets.
- Exploits collective features of MVAPICH2-GDR





More details in the HiDL Tutorial presented by Aamir Shafi and Nawras Alnassan yesterday

Outline

- Brief Overview of the MVAPICH Project
- Features of Recent Releases
- **Upcoming Features**
  - MVAPICH and MVAPICH-Plus 3.0 series
  - MVAPICH-Plus for AI-Accelerator Chips (Habana/Gaudi)
  - MVAPICH and OMB for FPGA
  - INAM
  - MPI4DL for Deep Learning
  - **MPI4Spark**
  - Conversational AI Interface (SAI)
- Conclusions

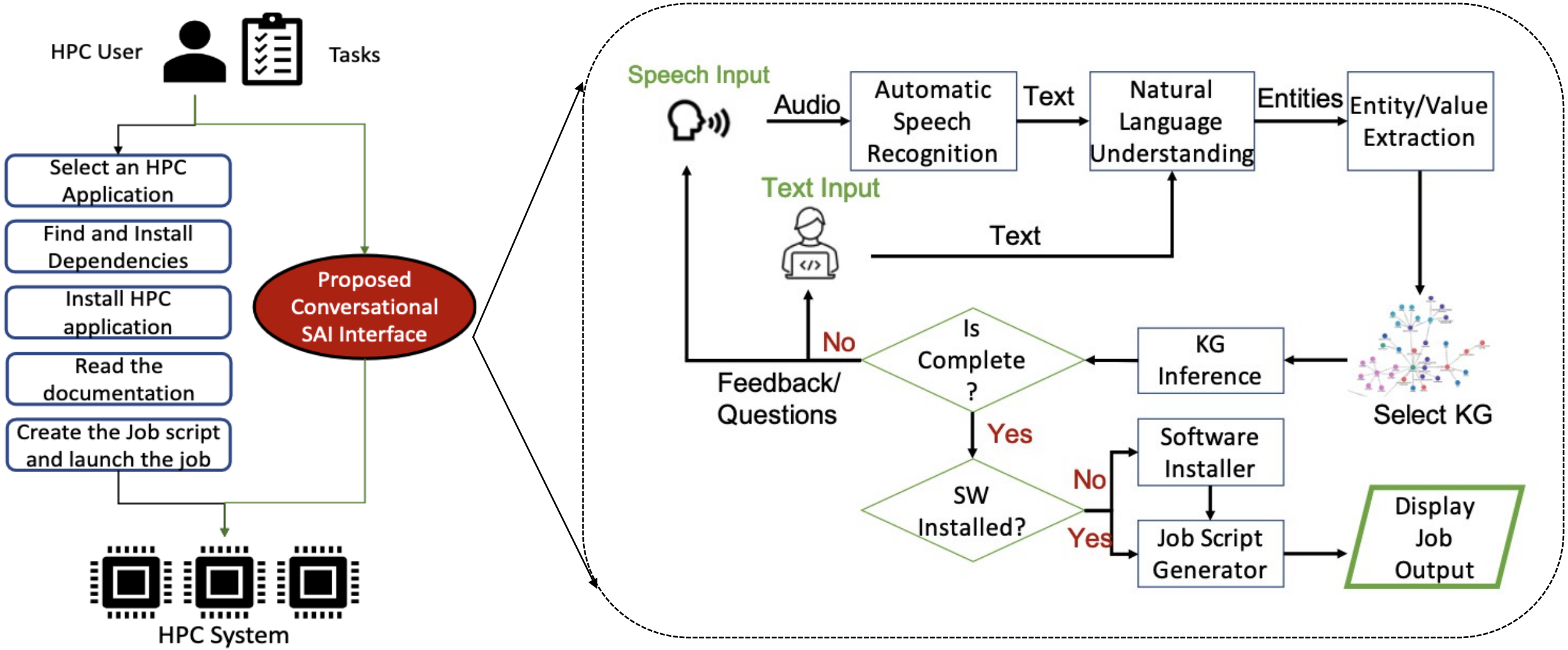
Plans for Next MPI4Spark Release (v0.2)

- MPI4Spark 0.2 will add support for the YARN resource manager to the MPI4Spark software
  - Will be available from: http://hibd.cse.ohio-state.edu
- Features:
  - (NEW) Support for the YARN Resource Manager
  - Based on Apache Spark 3.3.0
  - Compliant with user-level Apache Spark APIs and packages
  - High performance design that utilizes MPI-based communication
    - Utilizes MPI point-to-point operations
    - Relies on MPI Dynamic Process Management (DPM) features for launching executor processes
  - Built on top of the MVAPICH2-J Java bindings for MVAPICH2 family of MPI libraries
  - Tested with
    - (NEW) OSU HiBD-Benchmarks, GroupBy and SortBy
    - (NEW) Intel HiBench Suite, Micro Benchmarks, Machine Learning and Graph Workloads
    - Mellanox InfiniBand adapters (EDR and HDR 100G and 200G)
    - HPC systems with Intel OPA interconnects
    - Various multi-core platforms

Outline

- Brief Overview of the MVAPICH Project
- Features of Recent Releases
- **Upcoming Features**
  - MVAPICH and MVAPICH-Plus 3.0 series
  - MVAPICH-Plus for AI-Accelerator Chips (Habana/Gaudi)
  - MVAPICH and OMB for FPGA
  - INAM
  - MPI4DL for Deep Learning
  - MPI4Spark
  - **Conversational AI Interface (SAI)**
- Conclusions

Proposed Framework for Conversational AI for HPC Tasks



More details in the Short Talk, presented by Pouya Kousha (Yesterday 4:00-5:30 pm)

Outline

- Brief Overview of the MVAPICH Project
- Features of Recent Releases
- Upcoming Features
  - MVAPICH and MVAPICH-Plus 3.0 series
  - MVAPICH-Plus for AI-Accelerator Chips (Habana/Gaudi)
  - MVAPICH and OMB for FPGA
  - INAM
  - MPI4DL for Deep Learning
  - MPI4Spark
  - **Conversational AI Interface (SAI)**
- **Conclusions**

MVAPICH2 enabling life-changing NASA's DART mission

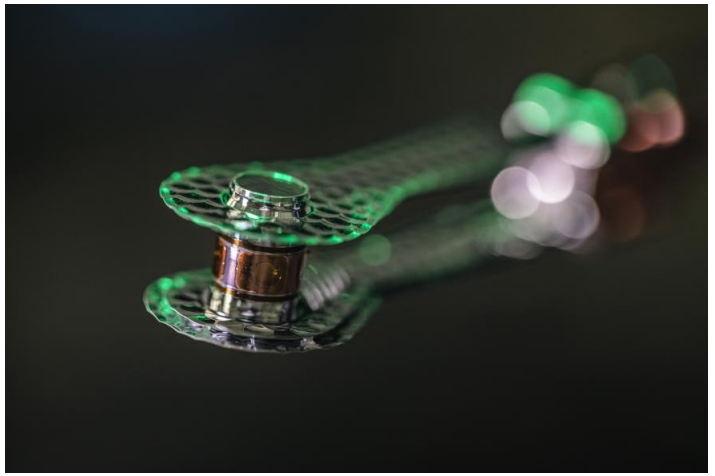
- Near-Earth asteroids (NEAs) have caused recent and ancient global catastrophes
  - LLNL scientists research ways to prevent NEAs using methods known as asteroid deflection
  - Joint NASA-LLNL research modelled various asteroid deflection methods (NASA's DART mission)
- MVAPICH2 lived at the core of the (NASA-DART mission) and enabled scalability
  - Underneath large-scale hydrodynamical and gravitational simulations required to compute the impact such as Spheral models

- https://twitter.com/NASA/status/1574539270987173903?s=20&t=u_4wIV9Cui2xyn9QLj286Q
- https://www.cbsnews.com/sanfrancisco/news/i-just-could-not-believe-it-livermore-team-celebrates-nasas-historic-strike-on-distant-asteroid/
- http://mug.mvapich.cse.ohio-state.edu/static/media/mug/presentations/18/moody-mug-18.pdf

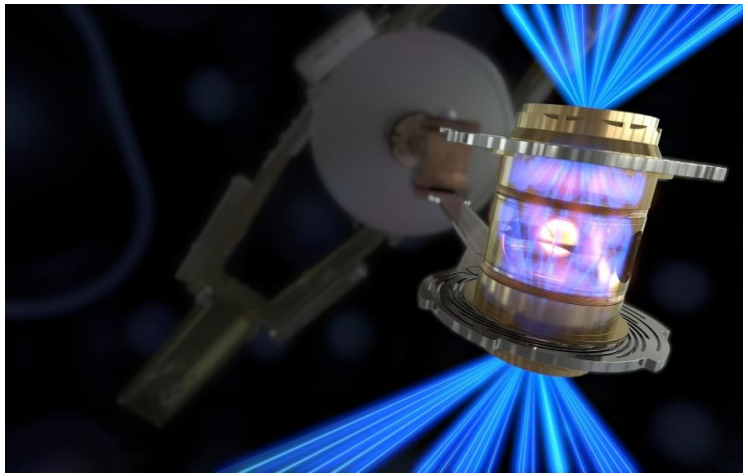
MVAPICH enabling Nuclear Fusion Research

- LLNL's National Ignition Facility (NIF) conducted the first controlled fusion experiment in history![1]
- MVAPICH, being the default MPI library on the LLNL systems, has been enabling the thousands of simulation jobs that have led to this amazing achievement!
- [1] https://www.llnl.gov/news/national-ignition-facility-achieves-fusion-ignition

The target chamber of LLNL's National Ignition Facility, where 192 laser beams delivered more than 2 million joules of ultraviolet energy to a tiny fuel pellet to create fusion ignition on Dec. 5, 2022.

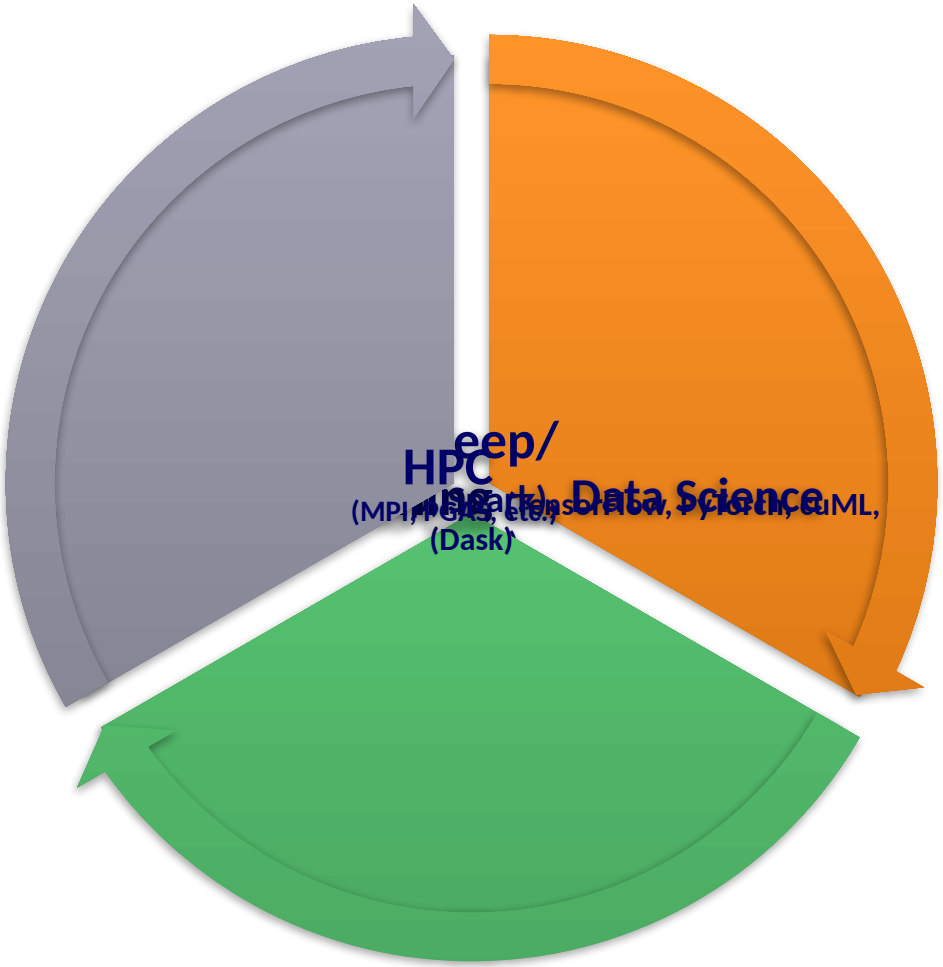


The hohlraum that houses the type of cryogenic target used to achieve ignition on Dec. 5, 2022, at LLNL's National Ignition Facility.



To create fusion ignition, the National Ignition Facility's laser energy is converted into X-rays inside the hohlraum, which then compress a fuel capsule until it implodes, creating a high temperature, high pressure plasma.

MPI-Driven Converged Software Stack for HPC, AI, Big Data and Data Science



Funding Acknowledgments

Funding Support by

Equipment Support by

Acknowledgments to all the Heroes (Past/Current Students and Staffs)

Current Students (Graduate)

- N. Alnaasan (Ph.D.)
- Q. Anthony (Ph.D.)
- C.-C. Chun (Ph.D.)
- N. Contini (Ph.D.)
- K. S. Khorassani (Ph.D.)
- P. Kousha (Ph.D.)
- B. Michalowicz (Ph.D.)
- B. Ramesh (Ph.D.)
- K. K. Suresh (Ph.D.)
- A. H. Tu (Ph.D.)
- S. Xu (Ph.D.)
- Q. Zhou (Ph.D.)
- K. Al Attar (M.S.)
- L. Xu (Ph.D.)
- G. Kuncham (Ph.D.)
- R. Vaidya (Ph.D.)
- J. Yao (P.hD.)
- M. Han (M.S.)
- A. Guptha (M.S.)

Current Research Scientists

- M. Abduljabbar
- A. Shafi

Current Students (Undergrads)

- J. Sulewski
- T. Chen

Current Faculty

- H. Subramoni

Current Software Engineers

- N. Pavuk
- N. Shineman
- M. Lieber

Current Research Specialist

- R. Motlagh

Past Students

- A. Awan (Ph.D.)
- A. Augustine (M.S.)
- P. Balaji (Ph.D.)
- M. Bayatpour (Ph.D.)
- R. Biswas (M.S.)
- S. Bhagvat (M.S.)
- A. Bhat (M.S.)
- D. Buntinas (Ph.D.)
- L. Chai (Ph.D.)
- B. Chandrasekharan (M.S.)
- S. Chakraborthy (Ph.D.)
- N. Dandapanthula (M.S.)
- V. Dhanraj (M.S.)
- C.-H. Chu (Ph.D.)
- T. Gangadharappa (M.S.)
- K. Gopalakrishnan (M.S.)
- J. Hashmi (Ph.D.)
- W. Huang (Ph.D.)
- A. Jain (Ph.D.)
- W. Jiang (M.S.)
- J. Jose (Ph.D.)
- M. Kedia (M.S.)
- 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.)
- K. Raj (M.S.)
- R. Rajachandrasekar (Ph.D.)
- D. Shankar (Ph.D.)
- G. Santhanaraman (Ph.D.)
- N. Sarkauskas (B.S. and M.S)
- N. Senthil Kumar (M.S.)
- A. Singh (Ph.D.)
- J. Sridhar (M.S.)
- S. Srivastava (M.S.)
- S. Sur (Ph.D.)
- H. Subramoni (Ph.D.)
- K. Vaidyanathan (Ph.D.)
- A. Vishnu (Ph.D.)
- J. Wu (Ph.D.)
- W. Yu (Ph.D.)
- J. Zhang (Ph.D.)

Past Post-Docs

- D. Banerjee
- X. Besseron
- M. S. Ghazimirsaeed
- H.-W. Jin
- J. Lin
- M. Luo
- E. Mancini
- K. Manian
- S. Marcarelli
- A. Ruhela
- J. Vienne
- H. Wang

Past Senior Research Associate

- J. Hashmi

Past Programmers

- A. Reifsteck
- D. Bureddy
- J. Perkins
- B. Seeds

Past Research Specialist

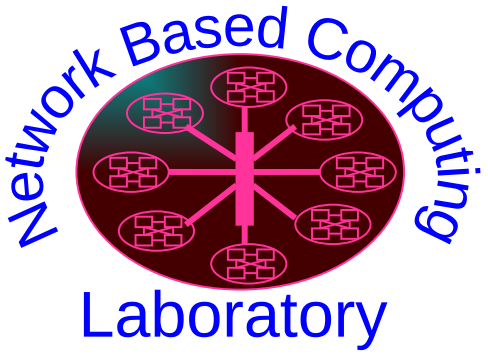
- M. Arnold
- J. Smith

Past Research Scientists

- K. Hamidouche
- S. Sur
- X. Lu

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/